

**TEST PROCEDURES**  
**For**  
**HIGH LEVEL ARCHITECTURE**  
**INTERFACE SPECIFICATION**

**May 1998**  
**VERSION 1.3**

Prepared by:

Margaret L. Loper  
Georgia Tech Research Institute  
Georgia Institute of Technology  
margaret.loper@gtri.gatech.edu

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 01-05-1998		2. REPORT TYPE		3. DATES COVERED (FROM - TO) xx-xx-1998 to xx-xx-1998	
4. TITLE AND SUBTITLE Test Procedures for High Level Architecture Interface Specification. Version 1.3 Unclassified				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Loper, Margaret L. ;				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME AND ADDRESS Georgia Tech Research Institute Georgia Institute of Technology Atlanta, GAxxxxx				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS United States Department of Defense Defense Modeling and Simulation Office 1901 N. Beauregard St., Suite 500 Alexandria, VA22311-1705				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APUBLIC RELEASE					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT This document contains test procedures for the High Level Architecture (HLA) Interface Specification, v1.3. Section 1 discusses the terminology used, the organization of the procedures, and how the procedures are defined.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:		17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19. NAME OF RESPONSIBLE PERSON	
		Public Release	75	Fenster, Lynn lfenster@dtic.mil	
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified		19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 703767-9007 DSN 427-9007	
				Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std Z39.18	

## TABLE OF CONTENTS

<b>1. OVERVIEW .....</b>	<b>5</b>
1.1 Definitions .....	5
1.2 How to Use the Test Procedures.....	5
1.3 Message Sequence Chart .....	6
1.3.1 <i>MSC Tutorial</i> .....	6
1.3.2 <i>MSCs and Testing</i> .....	7
<b>2. THE MASTER SEQUENCE.....</b>	<b>8</b>
<b>3. SERVICE SEQUENCES .....</b>	<b>10</b>
3.1 Federation Synchronization Sequences .....	10
3.1.1 <i>Register Synchronization</i> .....	10
3.1.1.1 Traceability .....	10
3.1.1.2 Service Set .....	11
3.1.2 <i>Respond to Synchronization</i> .....	11
3.1.2.1 Traceability .....	11
3.1.2.2 Service Set .....	11
3.2 Save and Restore Sequences.....	12
3.2.1 <i>Request Save Sequences</i> .....	12
3.2.1.1 Save Request.....	12
3.2.1.2 Restore Request .....	13
3.2.2 <i>Respond to Save Sequences</i> .....	13
3.2.2.1 Save Respond .....	14
3.2.2.2 Restore Respond .....	15
3.3 Object Sequences.....	15
3.3.1 <i>Request Attribute Value Update</i> .....	16
3.3.1.1 Traceability .....	16
3.3.1.2 Service Set .....	16
3.3.2 <i>Query Sequences</i> .....	16
3.3.2.1 Query and Inform Ownership.....	16
3.3.2.2 Is Attribute Owned by Federate.....	17
3.3.3 <i>Publish Object Attribute Sequences</i> .....	17
3.3.3.1 Register Object Sequence .....	18
3.3.3.2 Change Attribute Transport .....	21
3.3.3.3 Change Attribute Order .....	21
3.3.3.4 Publish Sequences .....	22
3.3.3.5 Update Sequences.....	24
3.3.3.6 Provide Attribute Update.....	27
3.3.3.7 Delete Object .....	28
3.3.4 <i>Ownership Management Sequences</i> .....	28
3.3.4.1 Assume Ownership.....	29
3.3.4.2 Divest Sequences .....	29
3.3.4.3 Request Release .....	33
3.3.4.4 Acquisition Sequences.....	34
3.3.5 <i>Subscribe Object Attribute Sequences</i> .....	38
3.3.5.1 Subscribe Object Sequences .....	39

3.3.5.2	Reflect Object Sequences .....	40
3.3.5.3	Remove Object .....	43
3.3.5.4	Local Delete Object .....	44
3.4	Interaction Sequences .....	44
3.4.1	<i>Change Interaction Transport</i> .....	45
3.4.1.1	Traceability .....	45
3.4.1.2	Service Set .....	45
3.4.2	<i>Change Interaction Order</i> .....	45
3.4.2.1	Traceability .....	46
3.4.2.2	Service Set .....	46
3.4.3	<i>Publish Interaction Sequences</i> .....	46
3.4.3.1	Publish Interaction .....	46
3.4.3.2	Unpublish Interaction .....	47
3.4.4	<i>Send Sequences</i> .....	48
3.4.4.1	Turn Interaction On and Off .....	48
3.4.4.2	Send Interaction .....	49
3.4.4.3	Disable Interaction Relevance Advisory .....	50
3.4.5	<i>Subscribe Interaction Sequences</i> .....	50
3.4.5.1	Subscribe Interaction .....	50
3.4.5.2	Unsubscribe Interaction .....	51
3.4.5.3	Receive Interaction .....	52
3.5	Time Sequences .....	52
3.5.1	<i>Regulation Sequences</i> .....	53
3.5.1.1	Enable Regulation .....	53
3.5.1.2	Disable Regulation .....	54
3.5.2	<i>Constrained Sequences</i> .....	54
3.5.2.1	Enable Constrained .....	55
3.5.2.2	Disable Constrained .....	55
3.5.3	<i>Delivery Sequences</i> .....	55
3.5.3.1	Enable Asynchronous Delivery .....	56
3.5.3.2	Disable Asynchronous Delivery .....	56
3.5.4	<i>Request Time Sequences</i> .....	57
3.5.4.1	Query Federate Time .....	57
3.5.4.2	Query LBTS .....	57
3.5.4.3	Query Minimum Next Event Time .....	58
3.5.5	<i>Lookahead Sequences</i> .....	58
3.5.5.1	Modify Lookahead .....	59
3.5.5.2	Query Lookahead .....	59
3.5.6	<i>Managed Time Sequences</i> .....	59
3.5.6.1	Time Advanced Request Sequences .....	60
3.5.6.2	Next Event Request Sequences .....	61
3.5.6.3	Flush Queue Request .....	63
3.5.7	<i>Retract Sequences</i> .....	63
3.5.7.1	Retract Sequence .....	63
3.5.7.2	Reflect Retract Sequences .....	65
3.6	Data Distribution Management Sequences .....	66

3.6.1	<i>Create Region</i> .....	66
3.6.1.1	Traceability .....	67
3.6.1.2	Service Set .....	67
3.6.2	<i>Modify Region</i> .....	67
3.6.2.1	Traceability .....	67
3.6.2.2	Service Set .....	67
3.6.3	<i>Delete Region</i> .....	67
3.6.3.1	Traceability .....	67
3.6.3.2	Service Set .....	67
3.6.4	<i>Register Object Sequences</i> .....	68
3.6.4.1	Associate Sequences .....	68
3.6.4.2	Unassociate Region for Updates .....	70
3.6.5	<i>Subscribe Object Region Sequences</i> .....	70
3.6.5.1	Subscribe Object Class Attributes with Region .....	71
3.6.5.2	Unsubscribe Object Class with Region .....	71
3.6.6	<i>Subscribe Interaction Region Sequences</i> .....	72
3.6.6.1	Subscribe Interaction Class with Region .....	72
3.6.6.2	Unsubscribe Interaction Class with Region .....	73
3.6.7	<i>Send Interaction with Region</i> .....	73
3.6.7.1	Traceability .....	73
3.6.7.2	Service Set .....	73
3.6.8	<i>Request Class Attribute Value Update with Region</i> .....	74
3.6.8.1	Traceability .....	74
3.6.8.2	Service Set .....	74
<b>4.</b>	<b>REFERENCES</b> .....	<b>75</b>

# 1. OVERVIEW

This document contains test procedures for the High Level Architecture (HLA) Interface Specification, v1.3. Section 1 discusses the terminology used, the organization of the procedures, and how the procedures are defined.

## 1.1 DEFINITIONS

The following definitions apply throughout this document:

Certification A federate is certified as HLA Compliant when it successfully passes all Conformance Tests administered by a Certification Agent.

Certification Agent (CA) The organization selected by DMSO to conduct Conformance Testing and issue the formal Certification of HLA Compliance. Currently, there is one CA approved by DMSO: AB Technologies.

Conformance Testing The process of verifying that a system under test performs in accordance with a standard [4]. For HLA, Conformance Testing ensures that a federate performs in accordance with the Object Model Template (OMT), the Interface Specification (IF Spec), and the HLA Rules.

FED (.fed) File The FED file is a representation of a federate's Simulation Object Model (SOM) or Federation Object Model (FOM). It specifies object classes and interactions, along with attributes and parameters, and it is formatted to show inheritance between classes.

Federate Under Test (FUT) The federate containing the Implementation Under Test.

HLA Compliance Defined by the HLA Compliance Checklist, which uses the following three documents to test for compliance: the OMT Specification, the IF Spec, and the HLA Rules [1].

Master Sequence A dependency tree of all services in the IF Spec, based on service prerequisites and post conditions.

Simulation Object Model (SOM) A specification of the intrinsic capabilities that an individual simulation offers to federations.

Test Sequence A test scenario that the FUT must execute to prove it can invoke and respond to the services specified in its Nominal and Representative SOM (RepSOM) test data [2]. The FUT receives a Test Sequence from the Certification Agent after it successfully passes the SOM Conformance Test and Conformance Cross-Check.

## 1.2 HOW TO USE THE TEST PROCEDURES

In accordance with the federate Compliance Checklist (items 2-6) [1], a federate must be capable of supporting the services in the Interface Specification. A federate does not

have to support all interface services, but it must indicate the subset of services it can support by completing a Conformance Statement<sup>1</sup> [2].

During Interface Testing, the Certification Agent must determine whether the Federate Under Test (FUT) demonstrates the services asserted in its Conformance Statement. This task is accomplished by specifying a Test Sequence that the federate must execute. The Test Sequence is composed of a set of service sequences (sets of related services) that map to the FUT's Conformance Statement.

These test procedures define the testable service sequences used in federate Conformance Testing. The test procedures use a notation called Message Sequence Charts (MSCs) as a means to describe the service sequences. The names of the services in a service sequence follow the Interface Specification; however, the names of the service sequences described in this document were developed specifically for Federate Testing. As a result, these names represent logical capabilities used in testing and should not be considered replacements for services defined in the Interface Specification.

### ***1.3 MESSAGE SEQUENCE CHART***

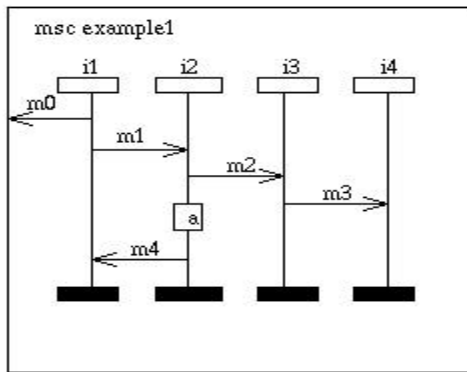
Message Sequence Charts (MSCs) represent a graphical and textual language that can be used to describe interactions between system components. The graphical form allows for an intuitive representation of the communications that occur during system execution. An MSC in graphical form is typical of diagrams used to describe communications protocols. Use of MSCs allows one to describe system behavior or specify use cases in distributed systems by focusing on the message interchange between communicating entities and their environment [3]. An MSC in proper graphical notation also corresponds unambiguously to a textual representation. Because of this property, it becomes convenient to manipulate MSCs and convey sequence information using the graphical notation. Conversely, it is more convenient to use the corresponding textual notation for processing MSC information computationally. The MSC concrete textual and graphical grammar is standardized as ITU (International Telecommunication Union) Recommendation Z.120.

#### **1.3.1 MSC Tutorial**

The Message Sequence Chart in Figure 1 defines the communication behavior between instances i1, i2, i3, and i4.

---

<sup>1</sup> A Conformance Statement is a list of the services a federate is capable of asserting and responding to.



**Figure 1**

An instance is denoted by a vertical axis. The time along each axis runs from top to bottom. A communication between two instances is represented by an arrow, which starts at the sending instance and ends at the receiving instance. In Figure 1, messages m1, m2, m3, and m4 are sent between instances, and message m0 is sent to the environment. The behavior of the environment is not specified. For instance i2, a

local action "a" was defined.

Although the activities along one single instance axis are completely ordered, there is no notion of global time. The only dependencies between the timing of the instances come from the restriction that a message must have been sent before it is received. In Figure 1, for example, this rule implies that message m3 is received by i4 only after it has been sent by i3, and, consequently, after the reception of m2 by i3. Thus, m1 and m3 are ordered in time, while no order is specified for m4 and m3. Since no particular communication mechanism is assumed, it would be possible to first send m3, then send and receive m4, and finally receive m3. The ordering of events on its own instance only restricts the execution of a local action.

### 1.3.2 MSCs and Testing

MSCs allow unambiguous specification of service sequences for testing. This feature permits the development of a test process that compares the service sequences observed during testing with the services stated in the Conformance Statement. This is accomplished by *matching* an observed set of services with the sequence described by an MSC. Conveniently, the matching operation can be performed using the synonymous textual representation.



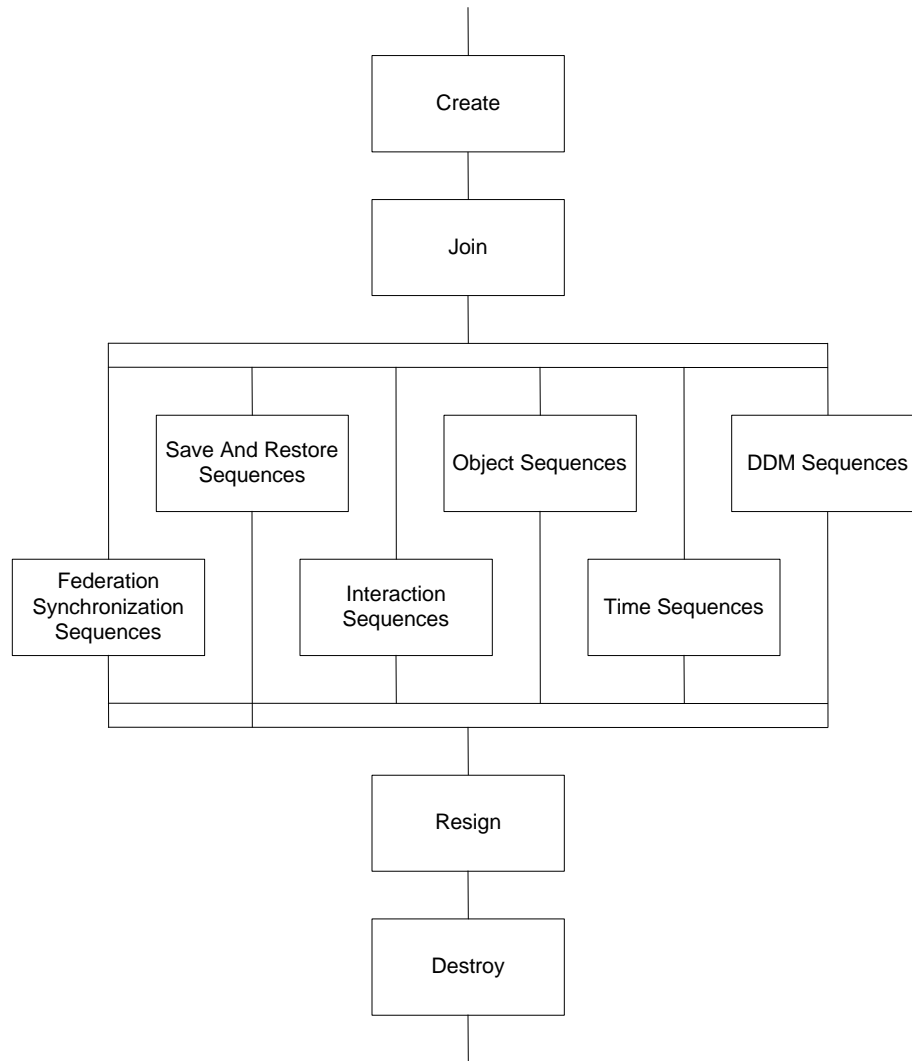
## 2. THE MASTER SEQUENCE

For Interface Conformance Testing, a federate must execute a Test Sequence that includes all the service sequences the federate specified it could support in its Conformance

Statement. The Test Sequence is derived from a *Master Sequence*, which is a dependency tree of all services in the Interface Specification, based on service prerequisites and post conditions. Where true dependencies exist (*e.g.*, Publish before Update), a service sequence indicates the mandatory ordering of related services. If no dependency exists (*e.g.*, Update and Save), the ordering of service sequences in the Master Sequence is arbitrary.

Based on the federate's Conformance Statement, the set of service sequences the federate is capable of invoking and responding to forms its "nominal sequence." If a federate can invoke and respond to all services in the Interface Specification, its Nominal Test Sequence is equivalent to the Master Sequence.

The Master Sequence is designed to support an arbitrary Conformance Statement (bounded, of course, by HLA Rules). In order to have a general statement of all logical sequences that can be specified in a Conformance Statement, the Master Sequence is organized hierarchically by service groupings. These groupings correspond to logical sets of capabilities for a federate, as shown in Figure 2. Subordinate sequences specify the ways in which the Interface Specification is used to accomplish an asserted capability.

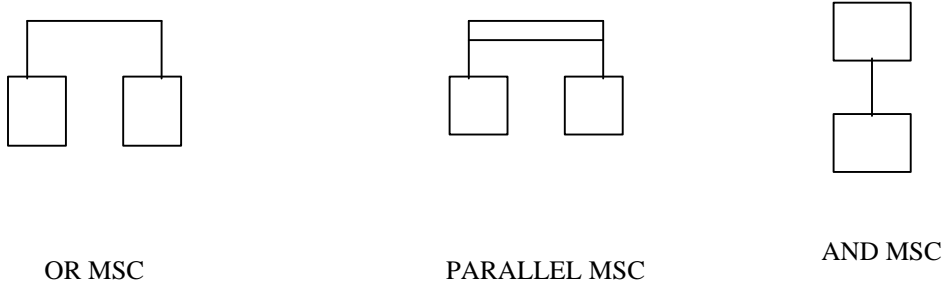


**Figure 2: Master Sequence MSC**

By organizing the Master Sequence in this way, we can create a mapping from each item in the Conformance Statement to a node in the Master Sequence tree. Thus, we can select a subset of the Master Sequence by "pruning" the irrelevant sub-trees. The resultant MSC is the Nominal Sequence.

### 3. SERVICE SEQUENCES

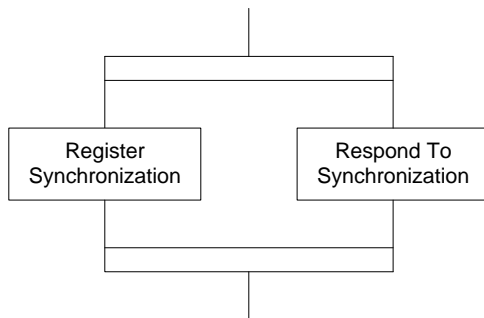
The following sections describe the service sequences contained in the Master Sequence. The charts that depict the service sequences (e.g., Figure 2) also describe the dependencies between them by using the notations *AND*, *OR*, and *PARALLEL*. The *AND* notation infers that each sub-sequence must occur in the order it appears. For the *OR* notation, the sub-sequences represent alternatives. The *PARALLEL* notation connotes that all sub-sequences contained in the sequence can be executed in any order, including concurrently. A graphical representation of these notations is shown in Figure 3.



**Figure 3. MSC Notations**

#### 3.1 FEDERATION SYNCHRONIZATION SEQUENCES

There are two types of synchronization sequences. The first type is when a federate requests a synchronization point for the federation. The second type is when a federate responds to a previously requested synchronization point. These service sequences are represented as *PARALLEL MSC* because several points may be registered at the same time. Thus, they are independent in their operation. The high-level MSC is shown below.



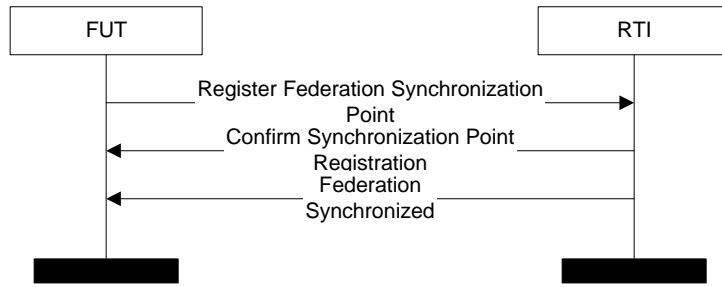
##### 3.1.1 Register Synchronization

This sequence represents the service invocations required for a federate to Register a Synchronization Point.

###### 3.1.1.1 Traceability

Section 4.6 Register Federation Synchronization Point, Section 4.7 Confirm Synchronization Point Registration †, and Section 4.10 Federation Synchronized.

### 3.1.1.2 Service Set



The *Register Federation Synchronization Point* service shall be used to initiate the registration of an upcoming synchronization point label.

The *Confirm Synchronization Point Registration* † service shall indicate to the federate the status of a requested federation synchronization point registration.

The *Federation Synchronized* † service shall inform the federate that all federates in the synchronization set of the specified synchronization point have invoked the *Synchronization Point Achieved* service for that point.

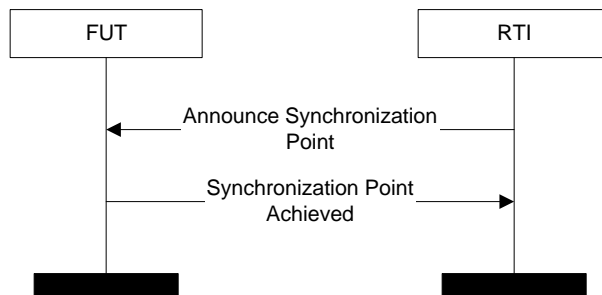
### 3.1.2 Respond to Synchronization

This sequence represents the service invocations required for a federate to Respond to a requested Synchronization Point.

#### 3.1.2.1 Traceability

Section 4.8 Announce Synchronization Point † and Section 4.9 Synchronization Point Achieved.

### 3.1.2.2 Service Set

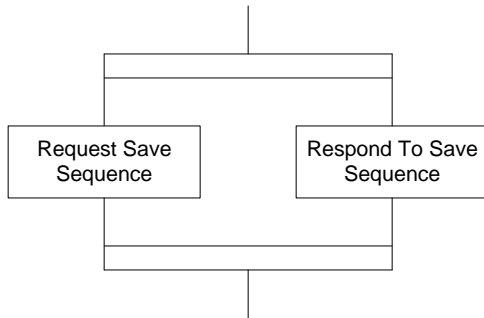


The *Announce Synchronization Point* † service shall inform a federate of the existence of a new synchronization point label.

The *Synchronization Point Achieved* service shall inform the RTI that the federate has reached the specified synchronization point.

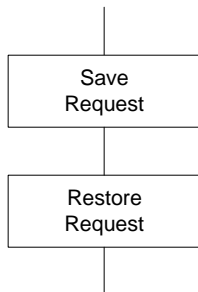
### 3.2 SAVE AND RESTORE SEQUENCES

The Save and Restore sequences are of two types. The first type is when a federate requests a Save and a Restore for the federation. The second type is when a federate responds to a requested Save and Restore. These service sequences are represented as a PARALLEL MSC as shown below.



#### 3.2.1 Request Save Sequences

The Request Save sequences represent the services required to Save and then Restore a federation. The service sequences are represented as an AND MSC. The reason is that a federate must request a Save before it can request a Restore. The high-level MSC is shown below.



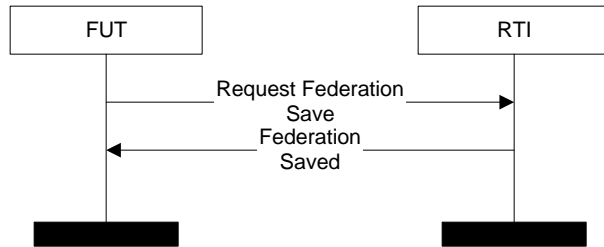
##### 3.2.1.1 Save Request

This sequence represents the service invocations required for a federate to Request a Federation Save.

###### 3.2.1.1.1 Traceability

Section 4.11 Request Federation Save and Section 4.15 Federation Saved †.

###### 3.2.1.1.2 Service Set



The *Request Federation Save* service shall specify that a federation save should take place.

The *Federation Saved* † service shall inform the federate that the federation save process is complete, and shall indicate whether it completed successfully or not.

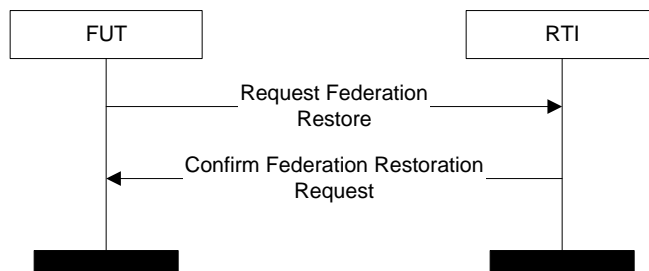
### 3.2.1.2 Restore Request

This sequence represents the service invocations required for a federate to Request a Federation Restore.

#### 3.2.1.2.1 Traceability

Section 4.16 Request Federation Restore and Section 4.17 Confirm Federation Restoration Request †.

#### 3.2.1.2.2 Service Set

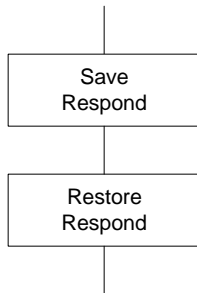


The *Request Federation Restore* service shall direct the RTI to begin the federation execution restoration process.

The *Confirm Federation Restoration Request* † shall indicate to the federate the status of a requested federation restoration.

### 3.2.2 Respond to Save Sequences

The Respond to Save sequences represent the services required to respond to a requested Save and Restore federation. The service sequences are represented as an AND MSC because a federate must respond to a requested Save before it can respond to a requested Restore. There is no limit on how many times a federate can respond to a Restore from a single Save request. The high-level MSC is shown below.



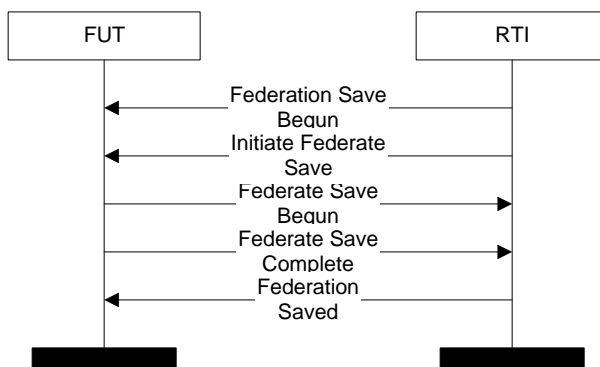
### 3.2.2.1 Save Respond

This sequence represents the service invocations required for a federate to Respond to a requested Federation Save.

#### 3.2.2.1.1 Traceability

Section 4.12 Initiate Federate Save †, Section 4.13 Federate Save Begun, Section 4.14 Federate Save Complete, and Section 4.15 Federation Saved †.

#### 3.2.2.1.2 Service Set



The *Initiate Federate Save* † service shall instruct the federate to save state.

The *Federate Save Begun* service shall notify the RTI that the federate is beginning to save its state.

The *Federate Save Complete* service shall notify the RTI that the federate has completed its save attempt. The save-success indicator shall inform the RTI that the federate save either succeeded or failed.

The *Federation Saved* † service shall inform the federate that the federation save process is complete, and shall indicate whether it completed successfully or not.

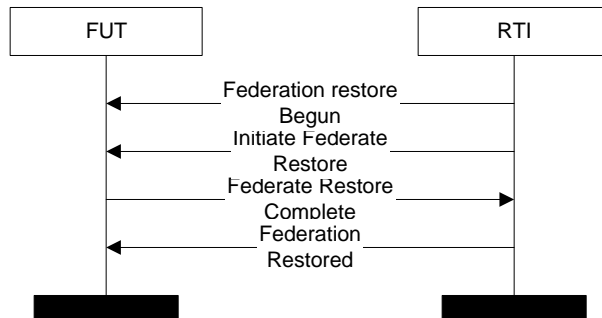
### 3.2.2.2 *Restore Respond*

This sequence represents the service invocations required for a federate to Respond to a requested Federation Restore.

#### 3.2.2.2.1 Traceability

Section 4.18 Federation Restore Begun †, Section 4.19 Initiate Federate Restore, Section 4.20 Federate Restore Complete, Section 4.21 Federation Restored †.

#### 3.2.2.2.2 Service Set



The *Federation Restore Begun* † service shall inform the federate that a federation restoration is imminent.

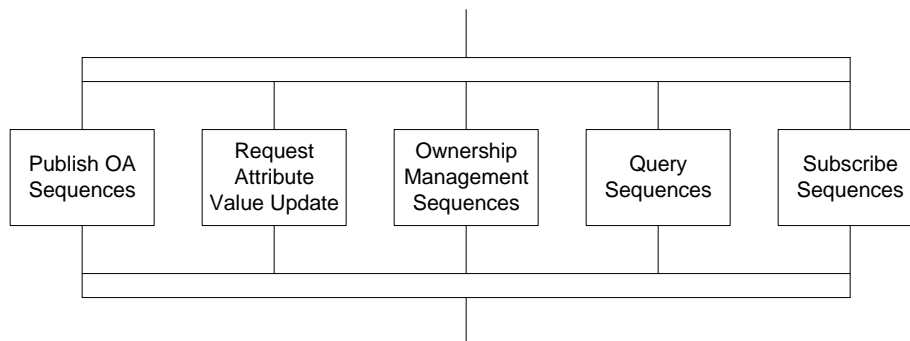
The *Initiate Federate Restore* † service shall instruct the federate to return to a previously saved state.

The *Federate Restore Complete* service shall notify the RTI that the federate has completed its restore attempt.

The *Federation Restored* † service shall inform the federate that the federation restore process is complete, and shall indicate whether it completed successfully or not.

### 3.3 OBJECT SEQUENCES

The Object Sequences represent the service sets related to objects and attributes. These service sequences are shown in the high-level MSC below.





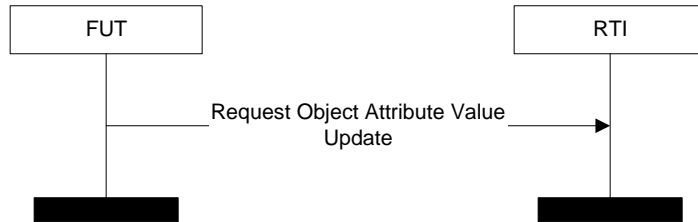
### 3.3.1 Request Attribute Value Update

This sequence represents the service invocation required for a federate to Request an Attribute Value Update.

#### 3.3.1.1 Traceability

Section 6.15 Request Attribute Value Update.

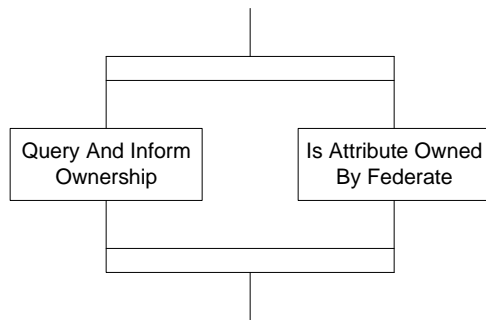
#### 3.3.1.2 Service Set



The *Request Attribute Value Update* service shall be used to stimulate the update of values of specified attributes.

### 3.3.2 Query Sequences

There are two types of Query sequences. The high-level MSc is shown below. These sequences are represented in a PARALLEL MSc since they can be done in any order, including concurrently.



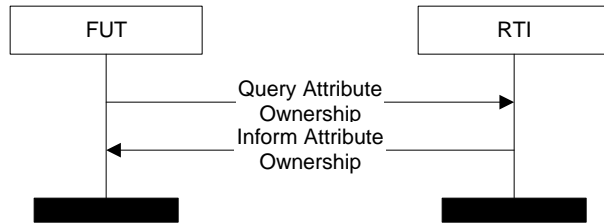
#### 3.3.2.1 Query and Inform Ownership

This sequence represents the service invocations required for a federate to Query Attribute Ownership.

##### 3.3.2.1.1 Traceability

Section 7.15 Query Attribute Ownership and Section 7.16 Inform Attribute Ownership †.

##### 3.3.2.1.2 Service Set



The *Query Attribute Ownership* service shall be used to determine the owner of the specified instance attribute.

The *Inform Attribute Ownership* † service shall be used to provide ownership information for the specified instance attribute.

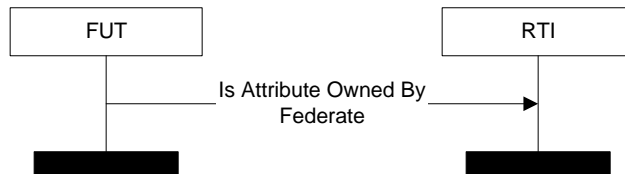
### 3.3.2.2 *Is Attribute Owned by Federate*

The sequence represents the service invocation required for a federate to query Is Attribute Owned by Federate.

#### 3.3.2.2.1 Traceability

Section 7.17 Is Attribute Owned by Federate.

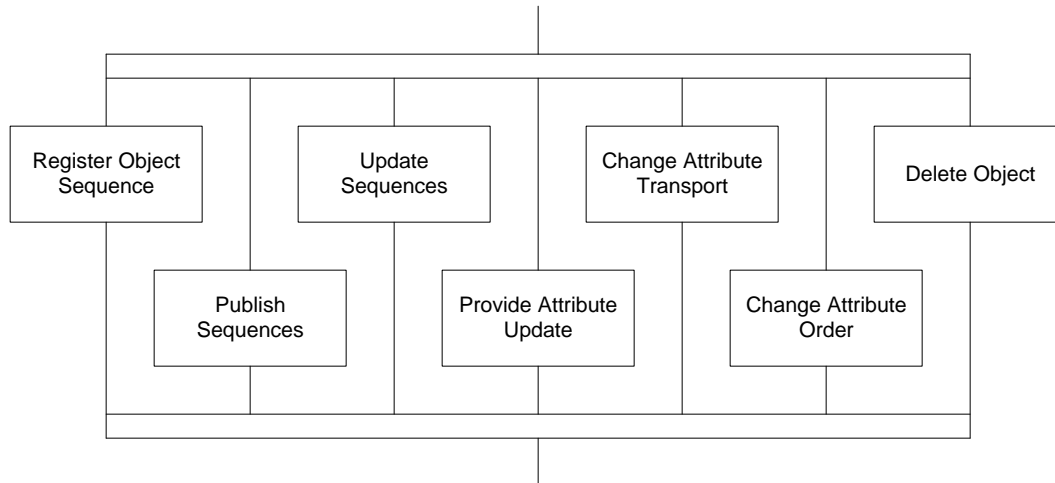
#### 3.3.2.2.2 Service Set



The *Is Attribute Owned By Federate* service shall be used to determine if the specified instance attribute of the specified object instance designator is owned by the invoking federate.

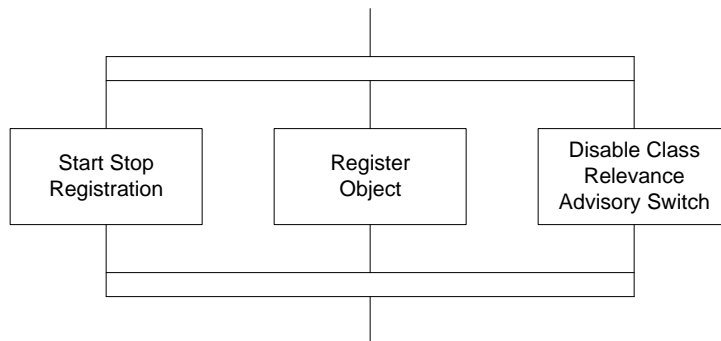
### 3.3.3 Publish Object Attribute Sequences

The Publish Object Attribute Sequences represent service sets associated with publishing object attributes. The high-level MSC is shown below. These sequences are represented as PARALLEL since they can be executed in any order.



### 3.3.3.1 Register Object Sequence

The Register Object sequences represent the service sets required to Register an Object. These sequences are shown below in a PARALLEL MSC since they can be executed in any order.



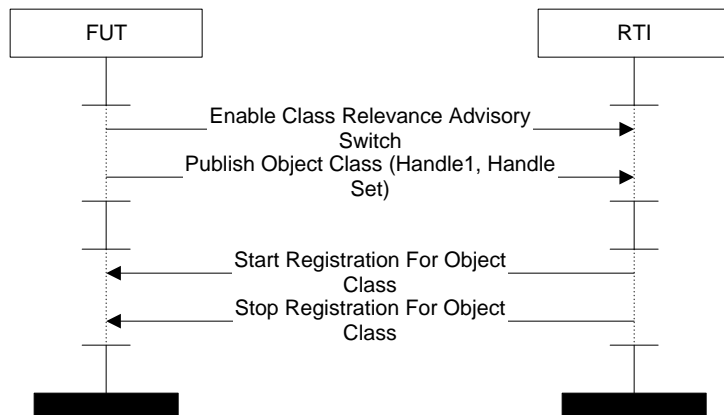
#### 3.3.3.1.1 Start and Stop Registration

This sequence represents the service invocations required for a federate to Enable Class Relevance and Start and Stop Registration. The hash marks along FUT and RTI axis in the MSC represent co-regions in which service invocations can be executed in any order. This service set is populated with SOM information for Federate testing.

##### 3.3.3.1.1.1 Traceability

Section 10.23 Enable Class Relevance Advisory Switch, Section 5.2 Publish Object Class, Section 5.10 Start Registration for Object Class †, and Section 5.11 Stop Registration for Object Class †.

##### 3.3.3.1.1.2 Service Set



The *Enable Class Relevance Advisory Switch* service shall set the Class Relevance Advisory switch on.

The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

#### Supplied Arguments

Object class designator

Set of attribute designators

The *Start Registration For Object Class* † service shall notify the federate that registration of new object instances of the specified object class is advised because at least one of the class attributes that the federate is publishing at this object class is actively subscribed to at the specified object class or at a super-class of the specified object class by at least one other federate in the federation execution.

The *Stop Registration For Object Class* † service shall notify the federate that registration of new object instances of the specified object class is not advised because none of the class attributes that the federate is publishing at this object class is actively subscribed to at the specified object class or at a super-class of the specified object class by any other federate in the federation execution.

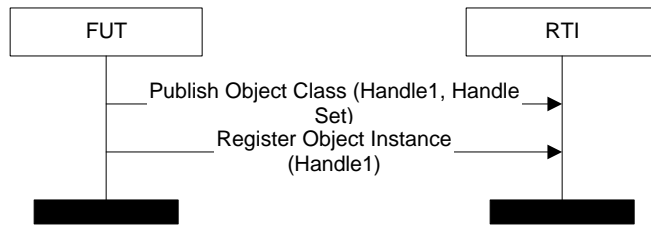
#### 3.3.3.1.2 Register Object

This sequence represents the service invocations required for a federate to Register Object Instance. This service set is populated with SOM information for Federate Testing.

##### 3.3.3.1.2.1 Traceability

Section 5.2 Publish Object Class and Section 6.2 Register Object Instance.

##### 3.3.3.1.2.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

#### Supplied Arguments

Object class designator

Set of attribute designators

#### Returned Arguments

None

The *Register Object Instance* service shall create a unique object instance designator and shall link it with an instance of the supplied object class.

#### Supplied Arguments

Object class designator

Optional object instance name

#### Returned Arguments

Object instance designator

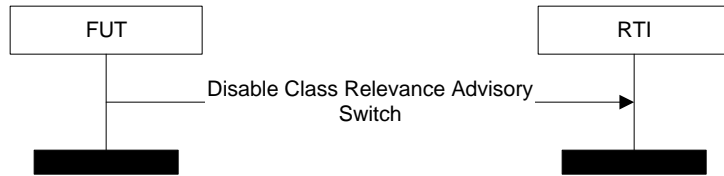
### 3.3.3.1.3 Disable Class Relevance Advisory Switch

This sequence represents the service invocations required for a federate to Disable Class Relevance Advisory.

#### 3.3.3.1.3.1 *Traceability*

Section 10.24 Disable Class Relevance Advisory Switch.

#### 3.3.3.1.3.2 *Service Set*



The *Disable Class Relevance Advisory Switch* service shall set the Class Relevance Advisory Switch off.

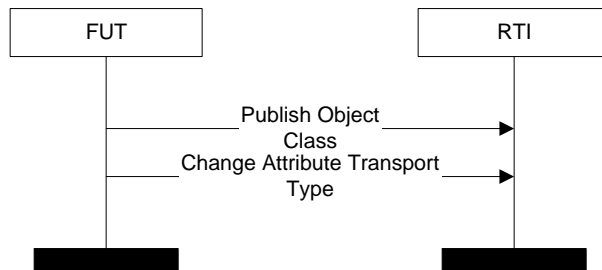
### 3.3.3.2 *Change Attribute Transport*

This sequence represents the service invocations required for a federate to Change Attribute Transportation Type.

#### 3.3.3.2.1 Traceability

Section 5.2 Publish Object Class and Section 6.11 Change Attribute Transportation Type.

#### 3.3.3.2.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

#### Supplied Arguments

Object class designator

Set of attribute designators

#### Returned Arguments

None

Invoking the *Change Attribute Transportation Type* service shall change the transportation type for all future *Update Attribute Values* service invocations for the specified attributes of the specified object instance only for the invoking federate.

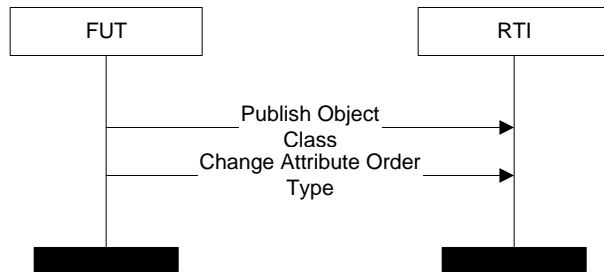
### 3.3.3.3 *Change Attribute Order*

This sequence represents the service invocations required for a federate to Change Attribute Order Type.

#### 3.3.3.3.1 Traceability

Section 5.2 Publish Object Class and Section 8.23 Change Attribute Order Type.

#### 3.3.3.3.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

##### Supplied Arguments

Object class designator

Set of attribute designators

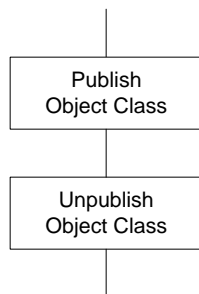
##### Returned Arguments

None

Invoking the *Change Attribute Order Type* service shall change the order type for all future *Update Attribute Values* service invocations for the specified instance attributes. When the ownership of an instance attribute is changed, the preferred order type shall revert to that defined in the FED.

#### 3.3.3.4 Publish Sequences

The Publish Object sequences represent the service sets associated with Publishing and Unpublishing an Object Class. These sequences are shown below in an AND MSC since they must be executed in the specified order.



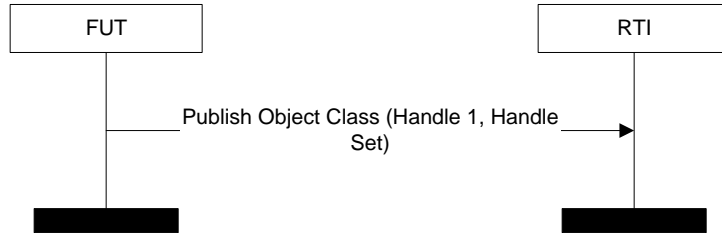
#### 3.3.3.4.1 Publish Object Class

This sequence represents the service invocations required for a federate to Publish Object Class. This service set is populated with SOM information for Federate Testing.

##### 3.3.3.4.1.1 Traceability

Section 5.2 Publish Object Class.

##### 3.3.3.4.1.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

##### Supplied Arguments

Object class designator

Set of attribute designators

##### Returned Arguments

None

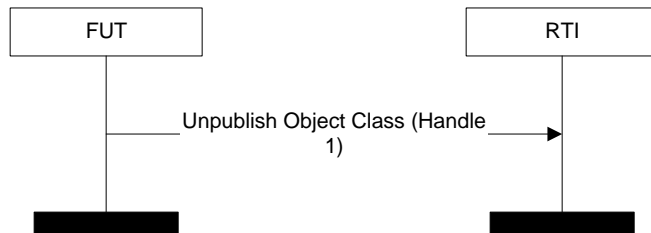
#### 3.3.3.4.2 Unpublish Object Class

This sequence represents the service invocations required for a federate to Unpublish Object Class. This service set is populated with SOM information for Federate Testing.

##### 3.3.3.4.2.1 Traceability

Section 5.3 Unpublish Object Class.

##### 3.3.3.4.2.2 Service Set





The *Unpublish Object Class* service shall inform the RTI that the federate will no longer register object instances of the specified object class.

Supplied Arguments

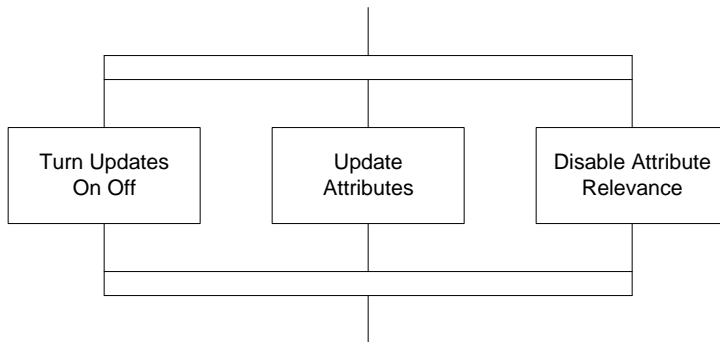
Object class designator

Returned Arguments

None

### 3.3.3.5 *Update Sequences*

The Update sequences represent the service sets associated with Updating Attribute Values. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be executed in any order.



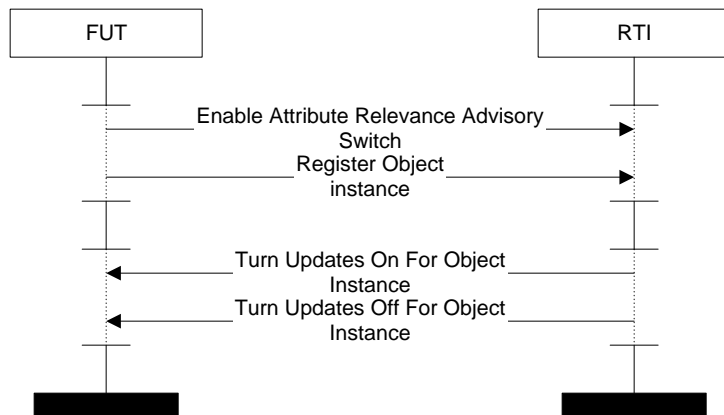
#### 3.3.3.5.1 *Turn Updates On and Off*

This sequence represents the service invocations required for a federate to Enable Attribute Relevance Advisory and Turn Updates On and Off for an Object Instance. The hash marks along FUT and RTI axis in the MSC represent co-regions in which the service invocations can be executed in any order.

##### 3.3.3.5.1.1 *Traceability*

Section 10.23 Enable Class Relevance Advisory Switch, Section 6.2 Register Object Instance, Section 5.12 Turn Interactions On †, and Section 5.13 Turn Interactions Off ‡.

##### 3.3.3.5.1.2 *Service Set*



The *Enable Class Relevance Advisory Switch* service shall set the Class Relevance Advisory Switch on.

The *Register Object Instance* service shall create a unique object instance designator and shall link it with an instance of the supplied object class.

The *Turn Interactions On*  $\dagger$  service shall notify the federate that the specified class of interactions is relevant because it or a super-class is actively subscribed to by at least one other federate in the federation execution.

The *Turn Interactions Off*  $\dagger$  service shall indicate to the federate that the specified class of interactions is not relevant because it or a super-class is not actively subscribed to by any other federate in the federation execution.

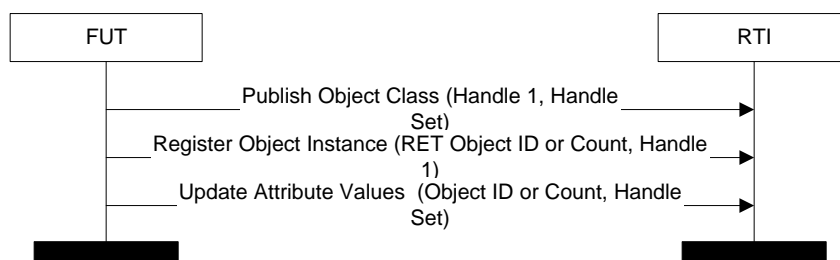
### 3.3.3.5.2 Update Attributes

This sequence represents the service invocations required for a federate to Update Attribute Values. This service set is populated with SOM information for Federate Testing.

#### 3.3.3.5.2.1 Traceability

Section 5.2 Publish Object Class, Section 6.2 Register Object Instance, and Section 6.4 Update Attribute Values.

#### 3.3.3.5.2.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

Supplied Arguments

Object class designator  
Set of attribute designators

Returned Arguments

None

The *Register Object Instance* service shall create a unique object instance designator and shall link it with an instance of the supplied object class.

Supplied Arguments

Object class designator  
Optional object instance name

Returned Arguments

Object instance designator

The *Update Attribute Values* service shall provide current values to the federation for instance attributes owned by the federate.

Supplied Arguments

Object instance designator  
Set of attribute designator and value pairs  
User-supplied tag  
Optional federation time

Returned Arguments

Optional event retraction designator

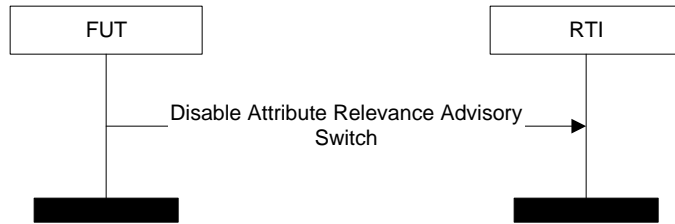
### 3.3.3.5.3 Disable Attribute Relevance

This sequence represents the service invocations required for a federate to Disable Attribute Relevance Advisory.

#### 3.3.3.5.3.1 Traceability

Section 10.26 Disable Attribute Relevance Advisory Switch.

#### 3.3.3.5.3.2 Service Set



The *Disable Attribute Relevance Advisory Switch* service shall set the Attribute Relevance Advisory switch off.

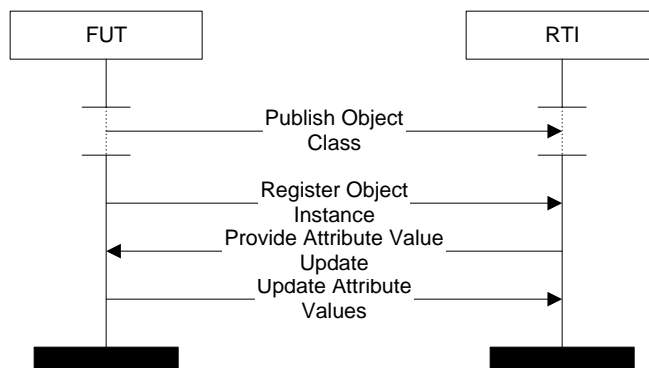
### 3.3.3.6 Provide Attribute Update

This sequence represents the service invocations required for a federate to Provide Attribute Update.

#### 3.3.3.6.1 Traceability

Section 5.2 Publish Object Class, Section 6.2 Register Object Instance, Section 6.4 Update Attribute Values, and Section 6.16 Provide Attribute Value Update †.

#### 3.3.3.6.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

The RTI shall create a unique object instance designator and shall link it with an instance of the supplied object class

The *Provide Attribute Value Update* † service shall request the current values for attributes owned by the federate for a given object instance.

The *Update Attribute Values* service shall provide current values to the federation for instance attributes owned by the federate.

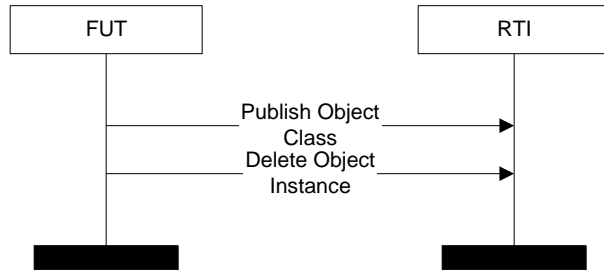
### 3.3.3.7 Delete Object

This sequence represents the service invocations required for a federate to Delete Object Instance.

#### 3.3.3.7.1 Traceability

Section 5.2 Publish Object Class and Section 6.8 Delete Object Instance.

#### 3.3.3.7.2 Service Set

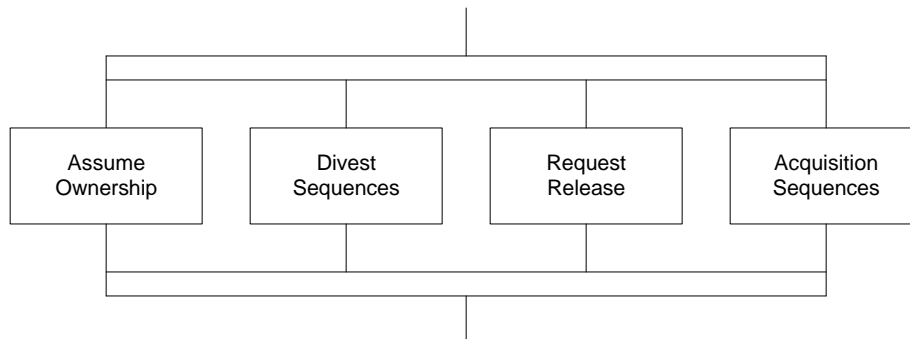


The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

The *Delete Object Instance* service shall inform the federation that an object instance with the specified designator, owned by the federate, is to be removed from the federation execution.

### 3.3.4 Ownership Management Sequences

The Ownership Management Sequences represent service associated with transferring object ownership. The high-level MSC is shown below. It is represented as PARALLEL since these sequences can be executed in any order.



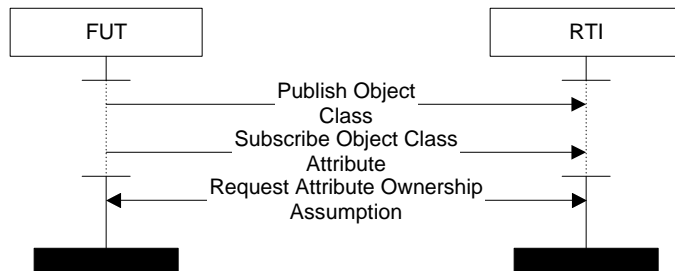
### 3.3.4.1 Assume Ownership

This sequence represents the service invocations required for a federate to Assume Ownership. The hash marks along FUT and RTI axis in the MSC represent co-regions in which the service invocations can be executed in any order.

#### 3.3.4.1.1 Traceability

Section 5.2 Publish Object Class, Section 5.6 Subscribe Object Class Attributes, and Section 7.4 Request Attribute Ownership Assumption †.

#### 3.3.4.1.2 Service Set



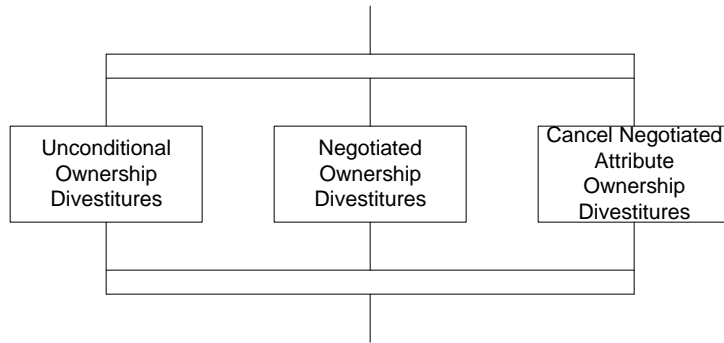
The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.

The *Request Attribute Ownership Assumption* † service shall inform the federate that the specified instance attributes are available for transfer of ownership to the federate

### 3.3.4.2 Divest Sequences

The Divest sequences represent the service sets required to Divest Ownership of object attributes. These sequences are shown below in a PARALLEL MSC since they can be executed in any order.



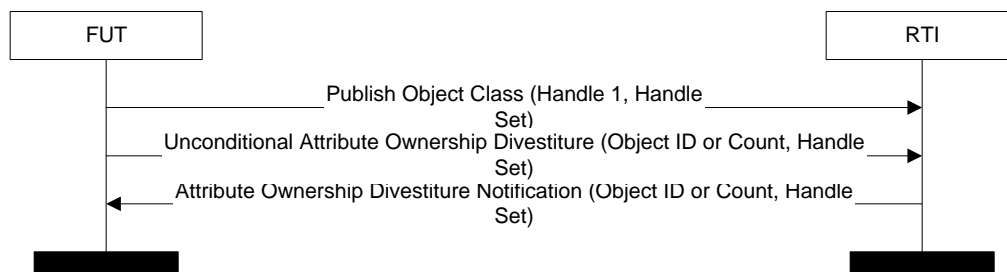
### 3.3.4.2.1 Unconditional Ownership Divestiture

This sequence represents the service invocations required for a federate to execute an Unconditional Ownership Divestiture. This service set is populated with SOM information for Federate testing.

#### 3.3.4.2.1.1 Traceability

Section 5.2 Publish Object Class, Section 7.2 Unconditional Attribute Ownership Divestiture, and Section 7.5 Attribute Ownership Divestiture Notification †.

#### 3.3.4.2.1.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

#### Supplied Arguments

Object class designator

Set of attribute designators

#### Returned Arguments

None

The *Unconditional Attribute Ownership Divestiture* service shall notify the RTI that the federate no longer wants to own the specified instance attributes of the specified object.

Supplied Arguments

Object instance designator  
Set of attribute designators

Returned Arguments

None

The *Attribute Ownership Divestiture Notification* † service shall notify the federate that it no longer owns the specified set of instance attributes.

Supplied Arguments

Object instance designator  
Set of attribute designators

Returned Arguments

None

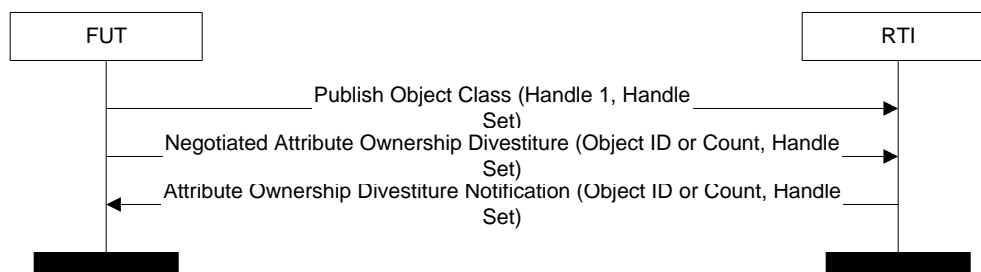
3.3.4.2.2 Negotiated Ownership Divestiture

This sequence represents the service invocations required for a federate to execute a Negotiated Ownership Divestiture. This service set is populated with SOM information for Federate Testing.

3.3.4.2.2.1 *Traceability*

Section 5.2 Publish Object Class, Section 7.3 Negotiated Attribute Ownership Divestiture, and Section 7.5 Attribute Ownership Divestiture Notification †.

3.3.4.2.2.2 *Service Set*



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

Supplied Arguments

Object class designator



Set of attribute designators

Returned Arguments

None

The *Negotiated Attribute Ownership Divestiture* service shall notify the RTI that the federate no longer wants to own the specified instance attributes of the specified object instance.

Supplied Arguments

Object instance designator

Set of attribute designators

User-supplied tag

Returned Arguments

None

The *Attribute Ownership Divestiture Notification* † service shall notify the federate that it no longer owns the specified set of instance attributes.

Supplied Arguments

Object instance designator

Set of attribute designators

Returned Arguments

None

### 3.3.4.2.3 Cancel Negotiated Attribute Ownership Divestiture

This sequence represents the service invocations required for a federate to execute a Cancel Negotiated Ownership Divestiture.

#### 3.3.4.2.3.1 Traceability

Section 7.12 Cancel Negotiated Attribute Ownership Divestiture.

#### 3.3.4.2.3.2 Service Set



The *Cancel Negotiated Attribute Ownership Divestiture* service shall notify the RTI that the federate no longer wants to divest ownership of the specified instance attributes.

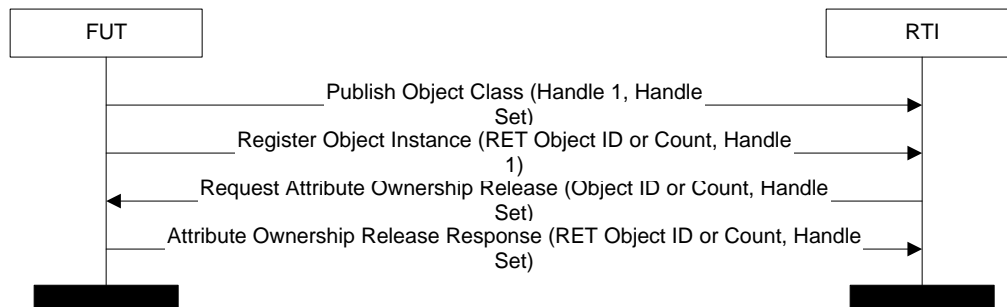
### 3.3.4.3 Request Release

This sequence represents the service invocations required for a federate to execute a Request Attribute Ownership Release. This service set is populated with SOM information for Federate Testing.

#### 3.3.4.3.1 Traceability

Section 5.2 Publish Object Class, Section 6.2 Register Object Instance, Section 7.10 Request Attribute Ownership Release †, and Section 7.11 Attribute Ownership Release Response.

#### 3.3.4.3.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

#### Supplied Arguments

Object class designator  
Set of attribute designators

#### Returned Arguments

None

The RTI shall create a unique object instance designator and shall link it with an instance of the supplied object class.

#### Supplied Arguments

Object class designator  
Optional object instance name

#### Returned Arguments

Object instance designator

The *Request Attribute Ownership Release* † service shall request that the federate release ownership of the specified instance attributes of the specified object instance.

Supplied Arguments

Object instance designator

Set of attribute designators

User-supplied tag

Returned Arguments

None

The *Attribute Ownership Release Response* service shall notify the RTI that the federate is willing to release ownership of the specified instance attributes for the specified object instance.

Supplied Arguments

Object instance designator

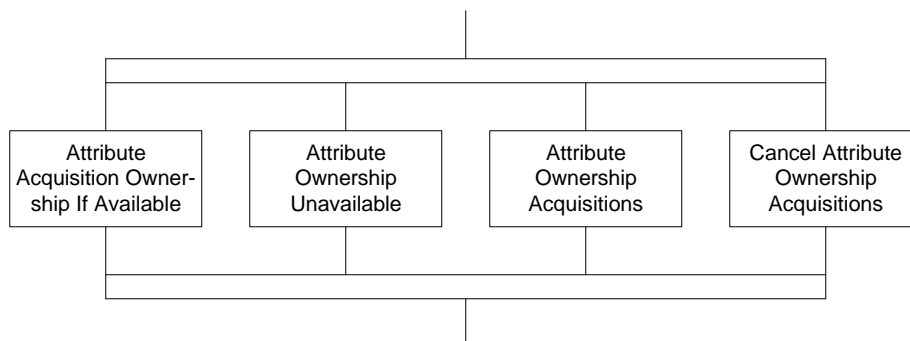
Set of attribute designators for which the federate is willing to release ownership

Returned Arguments

Set of attribute designators for which ownership is actually released

### 3.3.4.4 Acquisition Sequences

The Acquisition sequences represent the service sets required to Acquire Ownership of object attributes. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be executed in any order.



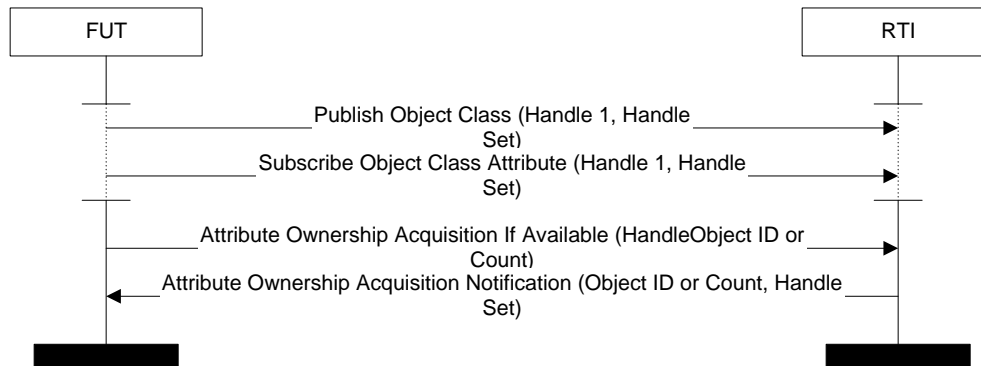
#### 3.3.4.4.1 Attribute Acquisition Ownership if Available

This sequence represents the service invocations required for a federate to execute an Attribute Acquisition Ownership if Available. The hash marks along FUT and RTI axis in the MSC represent co-regions in which the service invocations can be executed in any order. This service set is populated with SOM information for Federate Testing.

##### 3.3.4.4.1.1 Traceability

Section 5.2 Publish Object Class, Section 5.6 Subscribe Object Class Attributes, Section 7.8 Attribute Ownership Acquisition If Available, and Section 7.6 Attribute Ownership Acquisition Notification †.

#### 3.3.4.4.1.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

##### Supplied Arguments

Object class designator  
Set of attribute designators

##### Returned Arguments

None

The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.

##### Supplied Arguments

Object class designator  
Set of attribute designators  
Optional passive subscription indicator

##### Returned Arguments

None

The *Attribute Ownership Acquisition If Available* service shall request the ownership of the specified instance attributes of the specified object instance only if the instance attribute is unowned by all federates or it is in the process of being divested by its owner.

##### Supplied Arguments

Object instance designator  
Set of attribute designators

Returned Arguments

None

The *Attribute Ownership Acquisition Notification* † service shall notify the federate that it now owns the specified set of instance attributes.

Supplied Arguments

Object instance designator  
Set of attribute designators

Returned Arguments

None

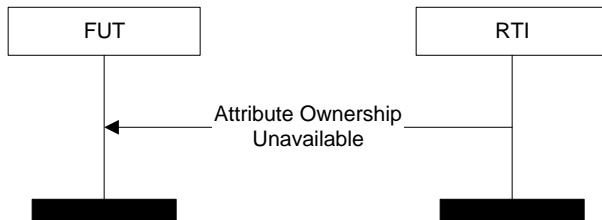
3.3.4.4.2 Attribute Ownership Unavailable

This sequence represents the service invocations required for a federate to receive an Attribute Ownership Unavailable.

3.3.4.4.2.1 Traceability

Section 7.9 Attribute Ownership Unavailable †.

3.3.4.4.2.2 Service Set



The *Attribute Ownership Unavailable* † service shall inform the federate that the specified instance attributes were not available for ownership acquisition.

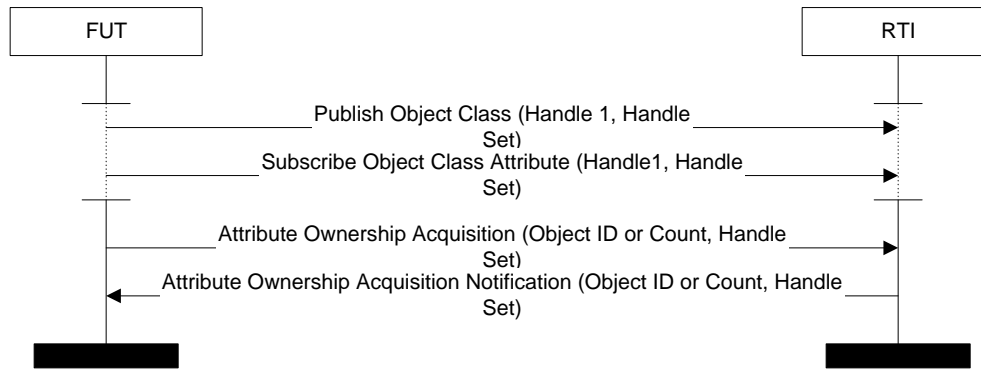
3.3.4.4.3 Attribute Ownership Acquisition

This sequence represents the service invocations required for a federate to execute Attribute Ownership Acquisition. The hash marks along FUT and RTI axis in the MSC represent co-regions in which the service invocations can be executed in any order. This service set is populated with SOM information for Federate Testing.

3.3.4.4.3.1 Traceability

Section 5.2 Publish Object Class, Section 5.6 Subscribe Object Class Attributes, Section 7.7 Attribute Ownership Acquisition, and Section 7.6 Attribute Ownership Acquisition Notification †.

#### 3.3.4.4.3.2 Service Set



The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

##### Supplied Arguments

Object class designator  
Set of attribute designators

##### Returned Arguments

None

The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.

##### Supplied Arguments

Object class designator  
Set of attribute designators  
Optional passive subscription indicator

##### Returned Arguments

None

The *Attribute Ownership Acquisition* service shall request the ownership of the specified instance attributes of the specified object instance.

##### Supplied Arguments

Object instance designator  
Set of attribute designators  
User-supplied tag

##### Returned Arguments

None

The *Attribute Ownership Acquisition Notification* † service shall notify the federate that it now owns the specified set of instance attributes.

Supplied Arguments

Object instance designator

Set of attribute designators

Returned Arguments

None

3.3.4.4.4 Cancel Attribute Ownership Acquisition

This sequence represents the service invocations required for a federate to execute a Cancel Attribute Ownership Acquisition.

3.3.4.4.4.1 Traceability

Section 7.13 Cancel Attribute Ownership Acquisition and Section 7.14 Confirm Attribute Ownership Acquisition Cancellation.

3.3.4.4.4.2 Service Set

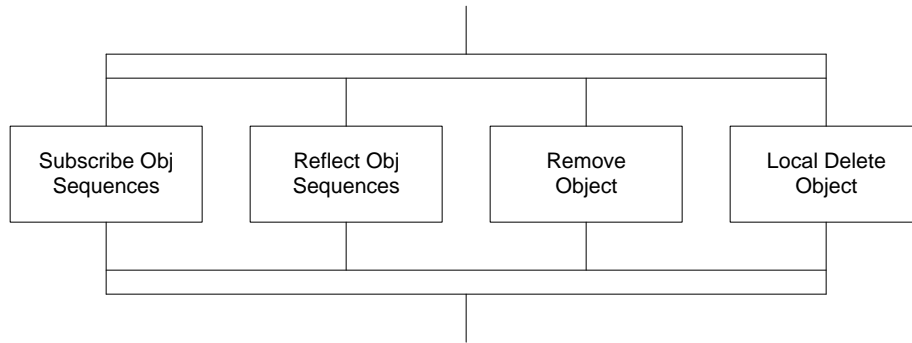


The *Cancel Attribute Ownership Acquisition* service shall notify the RTI that the federate no longer wants to acquire ownership of the specified instance attributes.

The *Confirm Attribute Ownership Acquisition Cancellation* † service shall inform the federate that the specified instance attributes are no longer candidates for ownership acquisition.

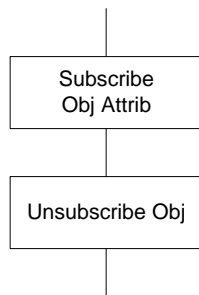
**3.3.5 Subscribe Object Attribute Sequences**

The Subscribe Object Attribute Sequences represent service sets associated with Subscribing to Object Attributes. The high-level MSC is shown below. It is represented as PARALLEL since these sequences can be executed in any order.



### 3.3.5.1 *Subscribe Object Sequences*

The Subscribe Object sequences represent the service sets required to Subscribe and Unsubscribe to an Object Class. These sequences are shown below in an AND MSC since they must be executed in the specified order.



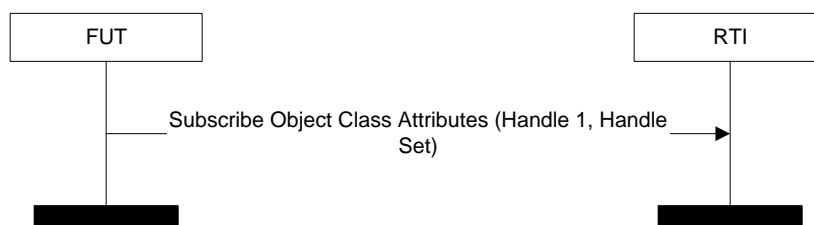
#### 3.3.5.1.1 *Subscribe Object Attribute*

This sequence represents the service invocations required for a federate to Subscribe Object Class. This service set is populated with SOM information for Federate Testing.

##### 3.3.5.1.1.1 *Traceability*

Section 5.6 Subscribe Object Class Attributes.

##### 3.3.5.1.1.2 *Service Set*



The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.



#### Supplied Arguments

Object class designator  
Set of attribute designators  
Optional passive subscription indicator

#### Returned Arguments

None

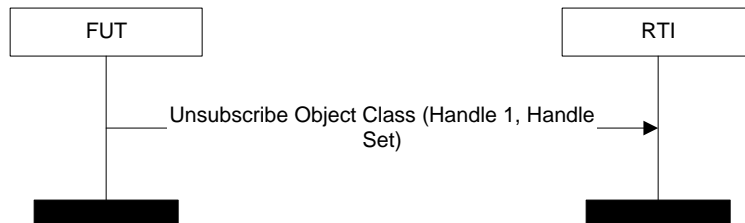
#### 3.3.5.1.2 Unsubscribe Object

This sequence represents the service invocations required for a federate to Unsubscribe Object Class. This service set is populated with SOM information for Federate Testing.

##### 3.3.5.1.2.1 *Traceability*

Section 5.7 Unsubscribe Object Class.

##### 3.3.5.1.2.2 *Service Set*



The *Unsubscribe Object Class* service shall inform the RTI that it is to stop notifying the federate of object instance discovery at the specified object class.

#### Supplied Arguments

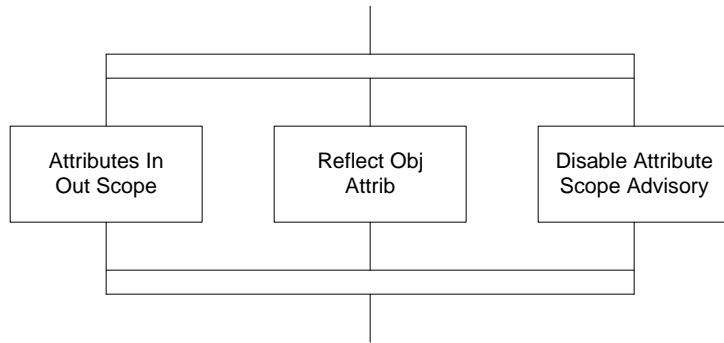
Object class designator

#### Returned Arguments

None

#### 3.3.5.2 *Reflect Object Sequences*

The Reflect Object sequences represent the service sets associated with Reflecting Attribute Values. These sequences are shown below in a PARALLEL MSC since they can be executed in any order.



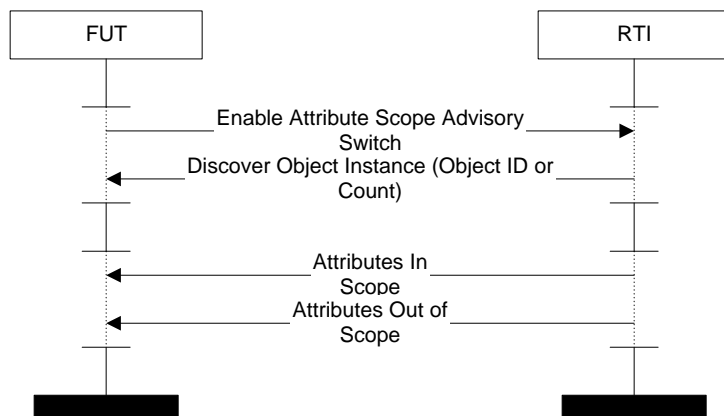
### 3.3.5.2.1 Attributes In or Out of Scope

This sequence represents the service invocations required for a federate to Enable Attribute Scope Advisory, Discover Object Instances, and turn Attributes In and Out of Scope. The hash marks along FUT and RTI axis in the MSC represent co-regions in which service invocations can be executed in any order. This service set is populated with SOM information for Federate Testing.

#### 3.3.5.2.1.1 Traceability

Section 10.27 Enable Attribute Scope Advisory Switch, Section 6.3 Discover Object Instance †, Section 6.13 Attributes In Scope †, and Section 6.14 Attributes Our Of Scope †.

#### 3.3.5.2.1.2 Service Set



The *Enable Attribute Scope Advisory Switch* service shall set the Attribute Scope Advisory switch on.

The *Discover Object Instance* † service shall inform the federate to discover an object instance.

#### Supplied Arguments

Object instance designator

Object class designator

Returned Arguments

None

The *Attributes In Scope* † service shall notify the federate that the specified attributes for the object instance are in scope for the federate.

The *Attributes Out Of Scope* † service shall notify the federate that the specified attributes of the object instance are out of scope for the federate.

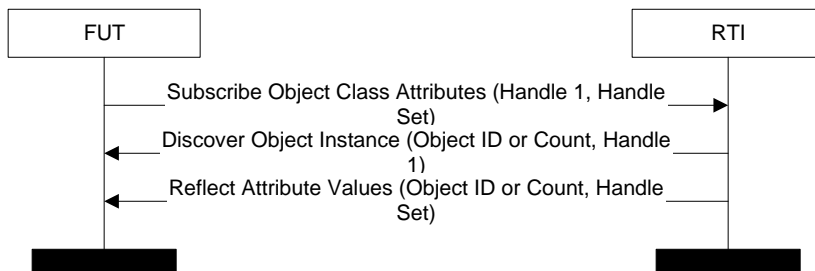
3.3.5.2.2 Reflect Object Attributes

This sequence represents the service invocations required for a federate to Reflect Object Attributes. This service set is populated with SOM information for Federate Testing.

3.3.5.2.2.1 Traceability

Section 5.6 Subscribe Object Class Attributes, Section 6.3 Discover Object Instance †, and Section 6.5 Reflect Attribute Values.

3.3.5.2.2.2 Service Set



The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.

Supplied Arguments

Object class designator

Set of attribute designators

Optional passive subscription indicator

Returned Arguments

None

The *Discover Object Instance* † service shall inform the federate to discover an object instance.

Supplied Arguments

Object instance designator

Object class designator

### Returned Arguments

None

The *Reflect Attribute Values* † service shall provide the federate with new values for the specified instance attributes.

### Supplied Arguments

Object instance designator  
Set of attribute designator and value pairs  
User-supplied tag  
Optional federation time  
Optional event retraction designator

### Returned Arguments

None

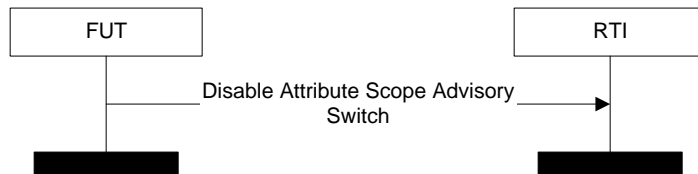
#### 3.3.5.2.3 Disable Attribute Scope Advisory Sequence

This sequence represents the service invocations required for a federate to Disable Attribute Scope Advisory.

##### 3.3.5.2.3.1 Traceability

Section 10.28 Disable Attribute Scope Advisory Switch.

##### 3.3.5.2.3.2 Service Set



The *Disable Attribute Scope Advisory Switch* service shall set the Attribute Scope Advisory switch off.

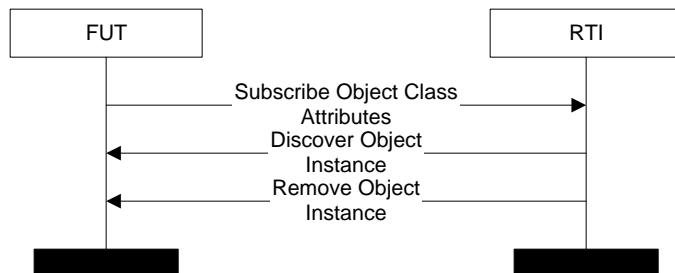
#### 3.3.5.3 Remove Object

This sequence represents the service invocations required for a federate to Remove Object.

##### 3.3.5.3.1 Traceability

Section 5.6 Subscribe Object Class Attributes, Section 6.3 Discover Object Instance †, and Section 6.9 Remove Object Instance †.

##### 3.3.5.3.2 Service Set



The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.

The *Discover Object Instance* † service shall inform the federate to discover an object instance.

The *Remove Object Instance* † service shall inform the federate that an object instance has been deleted from the federation execution.

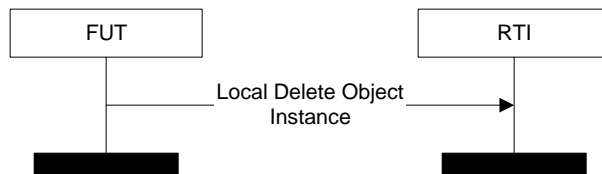
#### 3.3.5.4 Local Delete Object

This sequence represents the service invocations required for a federate to execute a Local Delete Object.

##### 3.3.5.4.1 Traceability

Section 6.10 Local Delete Object Instance.

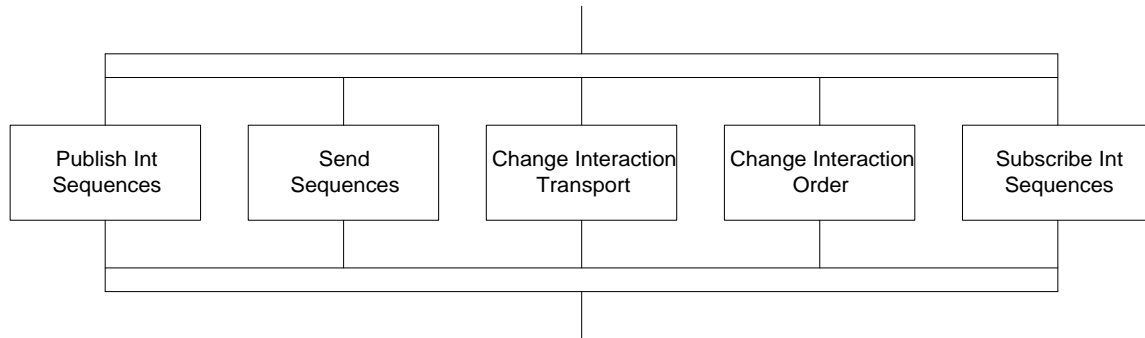
##### 3.3.5.4.2 Service Set



The *Local Delete Object Instance* service shall inform the RTI that it shall treat the specified object instance as if the RTI had never notified the invoking federate to discover the object instance.

### 3.4 INTERACTION SEQUENCES

The Interaction Sequences represent the service sets associated with Interactions. These service sequences are shown in the high-level MSC below.



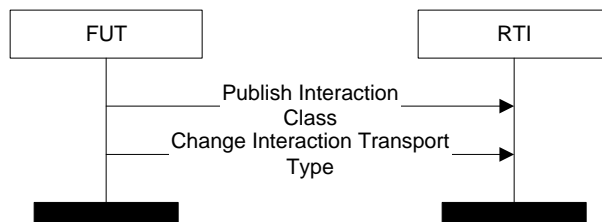
### 3.4.1 Change Interaction Transport

This sequence represents the service invocations required for a federate to Change Interaction Transportation Type.

#### 3.4.1.1 Traceability

Section 5.4 Publish Interaction Class and Section 6.12 Change Interaction Transportation Type.

#### 3.4.1.2 Service Set



The *Publish Interaction Class* service shall inform the RTI which classes of interactions the federate will send to the federation execution.

#### Supplied Arguments

Interaction class designator

#### Returned Arguments

None

Invoking the *Change Interaction Transportation Type* service shall change the transportation type for all future *Send Interaction* and *Send Interaction with Region* service invocations for the specified interaction class for the invoking federate only.

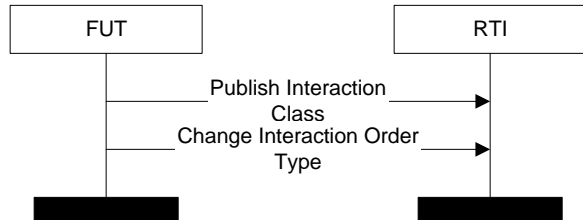
### 3.4.2 Change Interaction Order

This sequence represents the service invocations required for a federate to Change Interaction Order Type.

#### 3.4.2.1 Traceability

Section 5.4 Publish Interaction Class and Section 8.24 Change Interaction Order Type.

#### 3.4.2.2 Service Set



The *Publish Interaction Class* service shall inform the RTI which classes of interactions the federate will send to the federation execution.

##### Supplied Arguments

Interaction class designator

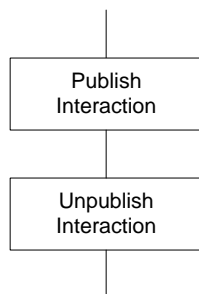
##### Returned Arguments

None

Invoking the *Change Interaction Order Type* service shall change the order type for all future *Send Interaction* and *Send Interaction with Region* service invocations for the specified interaction class for the invoking federate only.

#### 3.4.3 Publish Interaction Sequences

The Publish Interaction sequences represent the service sets required to Publish and Unpublish an Interaction Class. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.



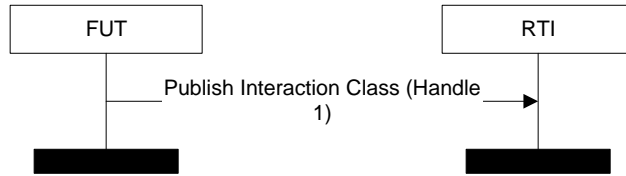
#### 3.4.3.1 Publish Interaction

This sequence represents the service invocations required for a federate to Publish Interaction Class. This service set is populated with SOM information for Federate Testing.

#### 3.4.3.1.1 Traceability

##### Section 5.4 Publish Interaction Class.

#### 3.4.3.1.2 Service Set



The *Publish Interaction Class* service shall inform the RTI which classes of interactions the federate will send to the federation execution.

#### Supplied Arguments

Interaction class designator

#### Returned Arguments

None

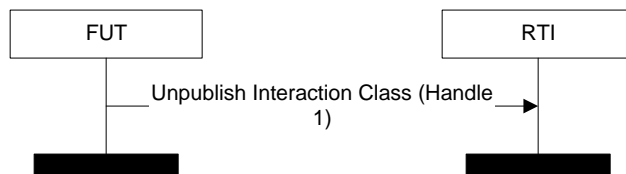
#### 3.4.3.2 *Unpublish Interaction*

This sequence represents the service invocations required for a federate to Unpublish Interaction Class. This service set is populated with SOM information for Federate Testing.

#### 3.4.3.2.1 Traceability

##### Section 5.5 Unpublish Interaction Class.

#### 3.4.3.2.2 Service Set



The *Unpublish Interaction Class* service shall inform the RTI that the federate will no longer send interactions of the specified class.

#### Supplied Arguments

Interaction class designator

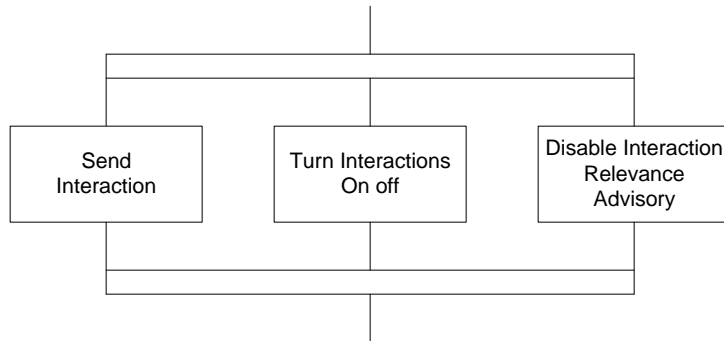
#### Returned Arguments

None



### 3.4.4 Send Sequences

The Send sequences represent the service sets associated with Sending Interactions. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be executed in any order.



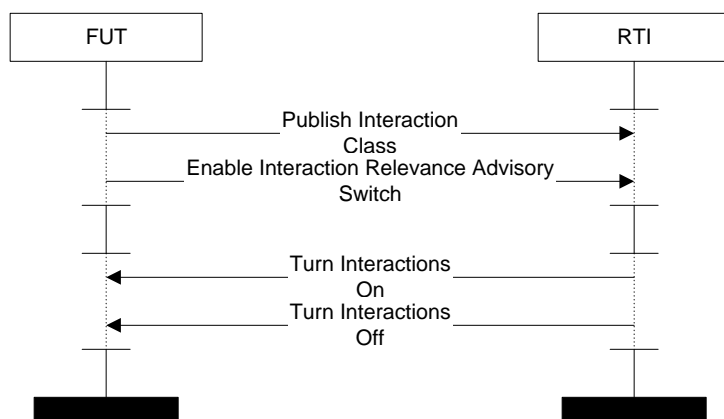
#### 3.4.4.1 Turn Interaction On and Off

This sequence represents the service invocations required for a federate to Enable Interaction Relevance Advisory and Turn Interactions On and Off for an Interaction Class. The hash marks along the FUT and RTI axes in the MSC represent co-regions in which services invocations can be executed in any order.

##### 3.4.4.1.1 Traceability

Section 5.4 Publish Interaction Class, Section 10.29 Enable Interaction Relevance Advisory Switch, Section 5.12 Turn Interactions On †, and Section 5.13 Interactions Off †.

##### 3.4.4.1.2 Service Set



The *Publish Interaction Class* service shall inform the RTI which classes of interactions the federate will send to the federation execution.

The *Enable Interaction Relevance Advisory Switch* service shall set the Interaction Relevance Advisory switch on.

The *Turn Interactions On* † service shall notify the federate that the specified class of interactions is relevant because it or a super-class is actively subscribed to by at least one other federate in the federation execution.

The *Turn Interactions Off* † service shall indicate to the federate that the specified class of interactions is not relevant because it or a super-class is not actively subscribed to by any other federate in the federation execution.

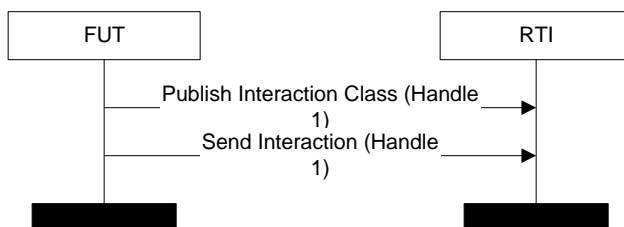
#### 3.4.4.2 *Send Interaction*

This sequence represents the service invocations required for a federate to Send Interaction. This service set is populated with SOM information for Federate Testing.

##### 3.4.4.2.1 Traceability

Section 5.4 Publish Interaction Class and Section 6.6 Send Interaction.

##### 3.4.4.2.2 Service Set



The *Publish Interaction Class* service shall inform the RTI which classes of interactions the federate will send to the federation execution.

#### Supplied Arguments

Interaction class designator

#### Returned Arguments

None

The *Send Interaction* service shall send an interaction into the federation.

#### Supplied Arguments

Interaction class designator

Set of interaction parameter designator and value pairs

User-supplied tag

Optional federation time

#### Returned Arguments

Optional event retraction designator

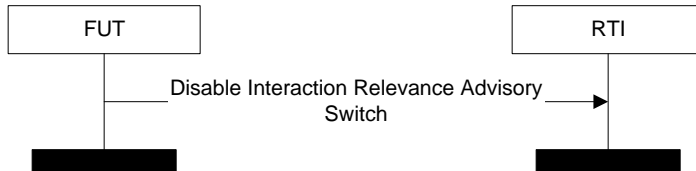
#### 3.4.4.3 *Disable Interaction Relevance Advisory*

This sequence represents the service invocations required for a federate to Disable Interaction Relevance Advisory.

##### 3.4.4.3.1 Traceability

Section 10.30 Disable Interaction Relevance Advisory Switch.

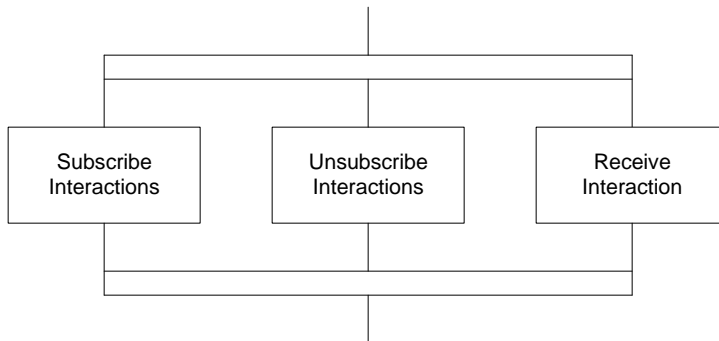
##### 3.4.4.3.2 Service Set



The *Disable Interaction Relevance Advisory Switch* service shall set the Interaction Relevance Advisory switch off.

#### 3.4.5 **Subscribe Interaction Sequences**

The Subscribe Interaction sequences represent the service sets associated with Subscribing to an Interaction Class. The high-level MSC is shown below. These sequences are represented in an AND (??) MSC since they must be executed in the specified order.



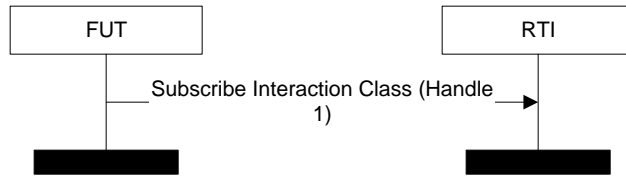
##### 3.4.5.1 *Subscribe Interaction*

This sequence represents the service invocations required for a federate to Subscribe Interaction Class. This service set is populated with SOM information for Federate Testing.

##### 3.4.5.1.1 Traceability

Section 5.8 Subscribe Interaction Class.

##### 3.4.5.1.2 Service Set



Specifies an interaction class for which the RTI should notify the federate of sent interactions by invoking the *Receive Interaction*  $\dagger$  service at the federate.

Supplied Arguments

Interaction class designator

Optional passive subscription indicator

Returned Arguments

None

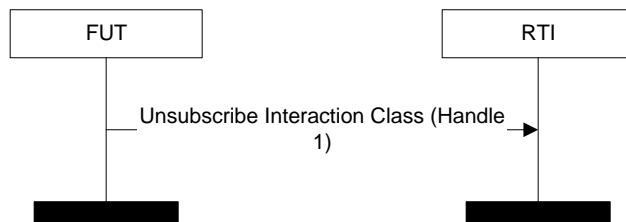
### 3.4.5.2 *Unsubscribe Interaction*

This sequence represents the service invocations required for a federate to Unsubscribe Interaction Class. This service set is populated with SOM information for Federate Testing.

#### 3.4.5.2.1 Traceability

Section 5.9 Unsubscribe Interaction Class.

#### 3.4.5.2.2 Service Set



The *Unsubscribe Interaction Class* service shall inform the RTI to no longer notify the federate of sent interactions of the specified interaction class.

Supplied Arguments

Interaction class designator

Returned Arguments

None

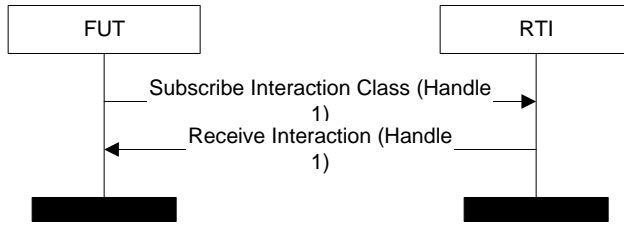
### 3.4.5.3 Receive Interaction

This sequence represents the service invocations required for a federate to Receive Interaction Class. This service set is populated with SOM information for Federate Testing.

#### 3.4.5.3.1 Traceability

Section 5.6 Subscribe Object Class Attributes, and Section 6.7 Receive Interaction †.

#### 3.4.5.3.2 Service Set



The *Subscribe Object Class Attributes* service shall specify an object class at which the RTI can notify the federate of discovery of object instances.

#### Supplied Arguments

Object class designator

Set of attribute designators

Optional passive subscription indicator

#### Returned Arguments

None

The *Receive Interaction* † service shall provide the federate with a sent interaction.

#### Supplied Arguments

Interaction class designator

Set of interaction parameter designator and value pairs

User-supplied tag

Optional federation time

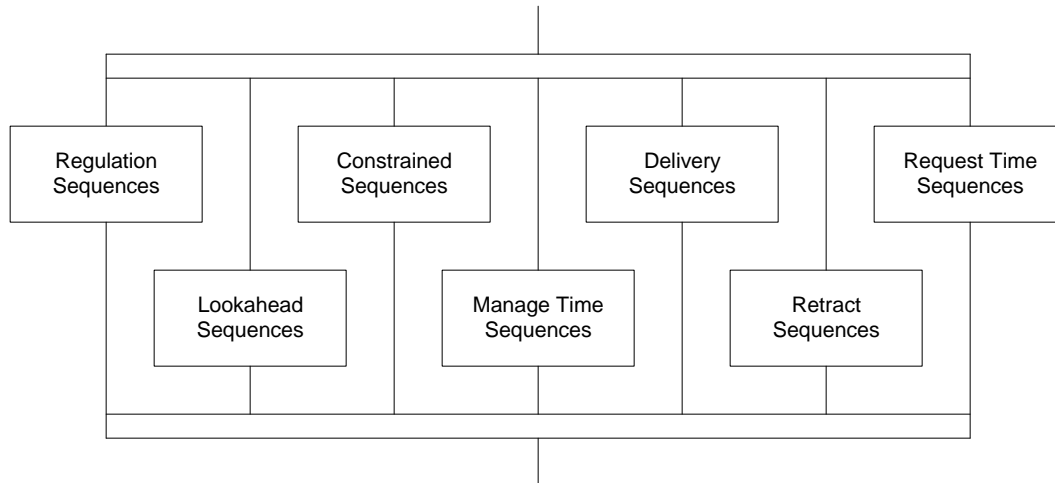
Optional event retraction designator

#### Returned Arguments

None

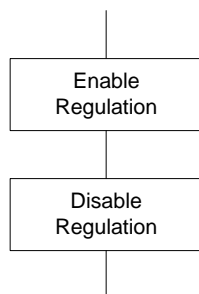
## 3.5 TIME SEQUENCES

The Time Sequences represent the service sets associated with Time management services. These service sequences are shown in the high-level MSC below.



### 3.5.1 Regulation Sequences

The Regulation sequences represent the service sets required to Enable and Disable Time Regulation. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.



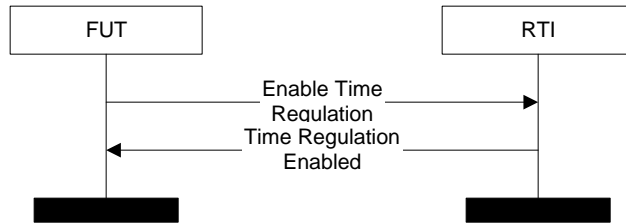
#### 3.5.1.1 *Enable Regulation*

This sequence represents the service invocations required for a federate to Enable Regulation.

##### 3.5.1.1.1 Traceability

Section 8.2 Enable Time Regulation and Section 8.3 Time Regulation Enabled †.

##### 3.5.1.1.2 Service Set



The *Enable Time Regulation* service shall enable time regulation for the federate invoking the service, thereby enabling the federate to send TSO messages.

Invocation of the *Time Regulation Enabled*  $\dagger$  service shall indicate that a prior request to enable time regulation has been honored.

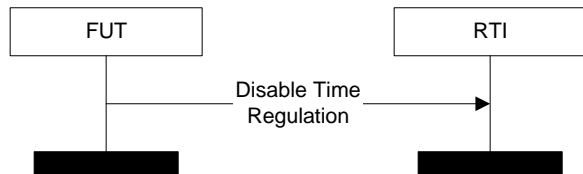
### 3.5.1.2 Disable Regulation

This sequence represents the service invocations required for a federate to Disable Regulation.

#### 3.5.1.2.1 Traceability

Section 8.4 Disable Time Regulation.

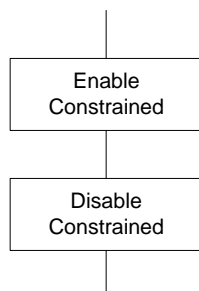
#### 3.5.1.2.2 Service Set



Invocation of the *Disable Time Regulation* service shall indicate that the federate is disabling time regulation.

## 3.5.2 Constrained Sequences

The Constrained sequences represent the service sets required to Enable and Disable Time Constrained. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.



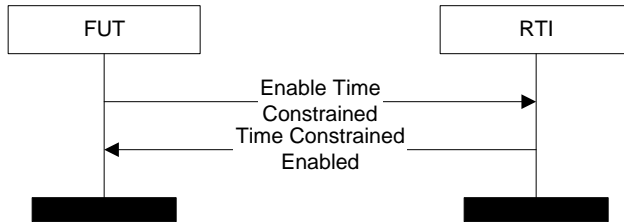
### 3.5.2.1 *Enable Constrained*

This sequence represents the service invocations required for a federate to Enable Time Constrained.

#### 3.5.2.1.1 Traceability

Section 8.5 Enable Time Constrained and Section 8.6 Time Constrained Enabled †.

#### 3.5.2.1.2 Service Set



The *Enable Time Constrained* service shall request that the federate invoking the service become time constrained.

Invocation of the *Time Constrained Enabled* † service shall indicate that a prior request to become time constrained has been honored.

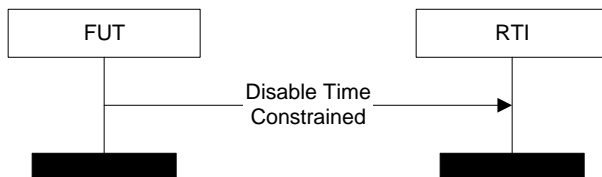
### 3.5.2.2 *Disable Constrained*

This sequence represents the service invocations required for a federate to Disable Time Constrained.

#### 3.5.2.2.1 Traceability

Section 8.7 Disable Time Constrained.

#### 3.5.2.2.2 Service Set

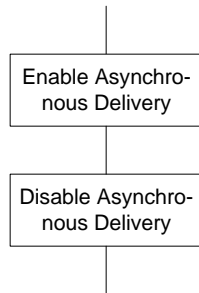


Invocation of the *Disable Time Constrained* service shall indicate that the federate is no longer time constrained.

## 3.5.3 Delivery Sequences

The Delivery sequences represent the service sets required to Enable and Disable Asynchronous Delivery. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.





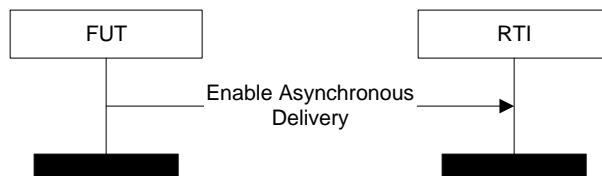
### 3.5.3.1 *Enable Asynchronous Delivery*

This sequence represents the service invocations required for a federate to Enable Asynchronous Delivery.

#### 3.5.3.1.1 Traceability

Section 8.14 Enable Asynchronous Delivery.

#### 3.5.3.1.2 Service Set



Invocations of the *Enable Asynchronous Delivery* service shall instruct the RTI to deliver received RO messages to the invoking federate when it is in either the Time Advancing or Time Granted state.

### 3.5.3.2 *Disable Asynchronous Delivery*

This sequence represents the service invocations required for a federate to Disable Asynchronous Delivery.

#### 3.5.3.2.1 Traceability

Section 8.15 Disable Asynchronous Delivery.

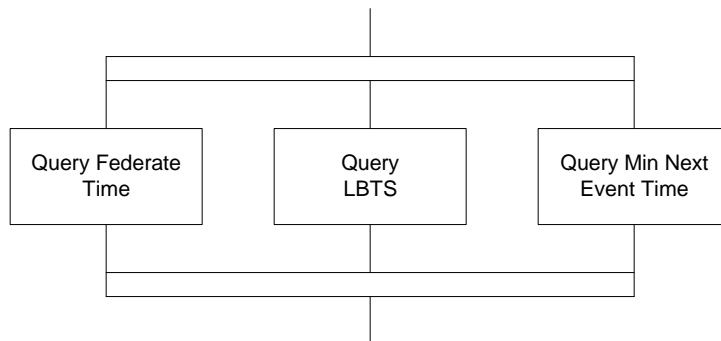
#### 3.5.3.2.2 Service Set



Invocations of the *Disable Asynchronous Delivery* service shall instruct the RTI to deliver received RO messages to the invoking federate only when it is in the Time Advancing state and the federate is time constrained.

### 3.5.4 Request Time Sequences

There are three types of Request Time sequences. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be done in any order, including concurrently.



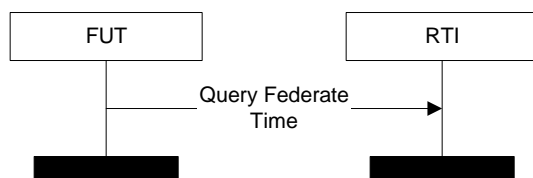
#### 3.5.4.1 Query Federate Time

This sequence represents the service invocations required for a federate to Query Federate Time.

##### 3.5.4.1.1 Traceability

Section 8.17 Query Federate Time.

##### 3.5.4.1.2 Service Set



The *Query Federate Time* service shall request the current value of the invoking federate's logical time.

#### 3.5.4.2 Query LBTS

The sequence represents the service invocation required for a federate to Query LBTS.

##### 3.5.4.2.1 Traceability

Section 8.16 Query LBTS.

##### 3.5.4.2.2 Service Set



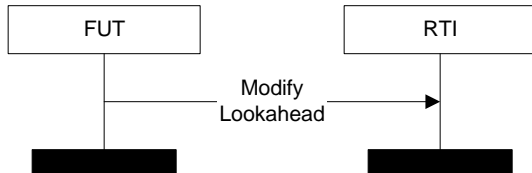
### 3.5.5.1 *Modify Lookahead*

This sequence represents the service invocations required for a federate to Modify Lookahead.

#### 3.5.5.1.1 Traceability

Section 8.19 Modify Lookahead.

#### 3.5.5.1.2 Service Set



The *Modify Lookahead* service shall request a change to the actual value of the federate's lookahead.

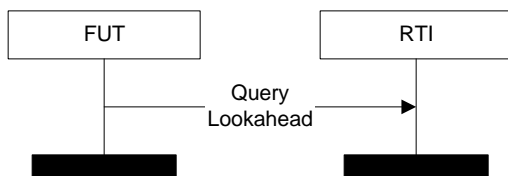
### 3.5.5.2 *Query Lookahead*

The sequence represents the service invocation required for a federate to Query Lookahead.

#### 3.5.5.2.1 Traceability

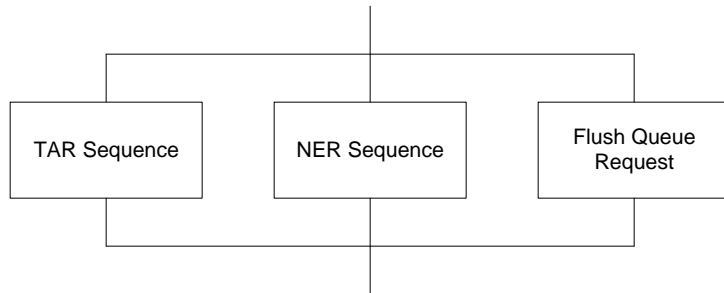
Section 8.20 Query Lookahead.

#### 3.5.5.2.2 Service Set



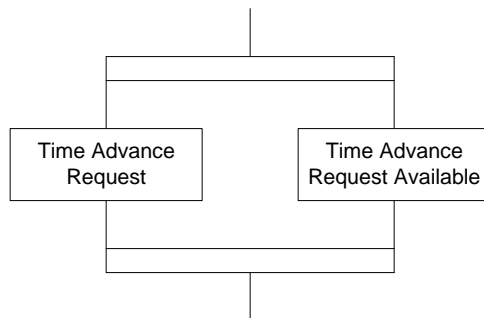
## 3.5.6 Managed Time Sequences

The Managed Time sequences represent the service sets associated with managing time in a federate. The high-level MSC is shown below. These sequences are represented in an OR MSC since one scheme will be selected for a federate.



### 3.5.6.1 Time Advanced Request Sequences

The Time Advance Request sequences represent the service sets associated with Time Advance Request. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be executed in any order.



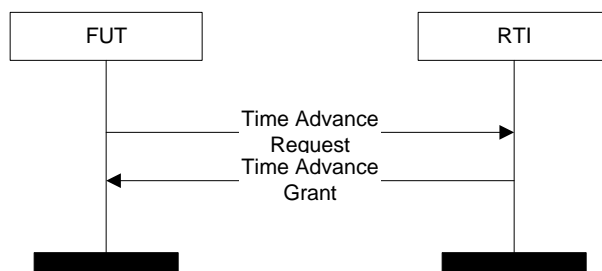
#### 3.5.6.1.1 Time Advance Request

This sequence represents the service invocations required for a federate to execute Time Advance Request.

##### 3.5.6.1.1.1 Traceability

Section 8.8 Time Advance Request and Section 8.13 Time Advance Grant †.

##### 3.5.6.1.1.2 Service Set



The *Time Advance Request* service shall request an advance of the federate's logical time and release zero or more messages for delivery to the federate.

Invocation of the *Time Advance Grant* † service shall indicate that a prior request to advance the federate's logical time has been honored.

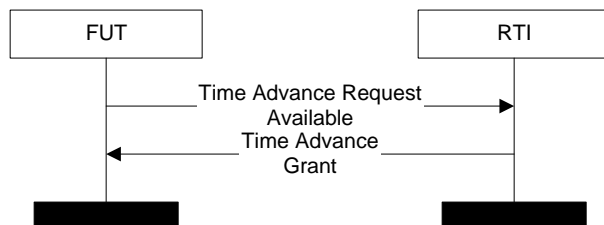
### 3.5.6.1.2 Time Advance Request Available

The sequence represents the service invocation required for a federate to execute Time Advance Request Available.

#### 3.5.6.1.2.1 Traceability

Section 8.9 Time Advance Request Available and Section 8.13 Time Advance Grant †.

#### 3.5.6.1.2.2 Service Set

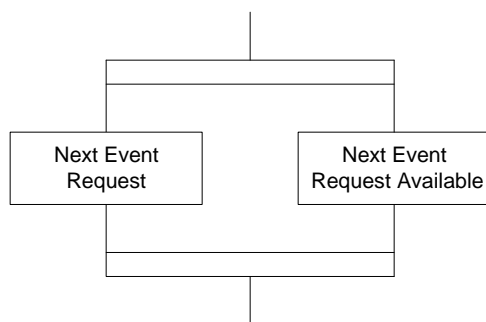


The *Time Advance Request Available* service shall request an advance of the federate's logical time.

Invocation of the *Time Advance Grant* † service shall indicate that a prior request to advance the federate's logical time has been honored.

### 3.5.6.2 Next Event Request Sequences

The Next Event Request sequences represent the service sets associated with Next Event Request. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be executed in any order.



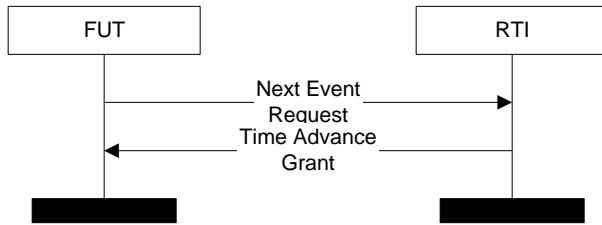
#### 3.5.6.2.1 Next Event Request

This sequence represents the service invocations required for a federate to execute Next Event Request.

##### 3.5.6.2.1.1 Traceability

Section 8.10 Next Event Request and Section 8.13 Time Advance Grant †.

#### 3.5.6.2.1.2 Service Set



The *Next Event Request* service shall request the logical time of the federate to be advanced to the time stamp of the next TSO message that will be delivered to the federate, provided that message has a time stamp no greater than the logical time specified in the request.

Invocation of the *Time Advance Grant* † service shall indicate that a prior request to advance the federate's logical time has been honored.

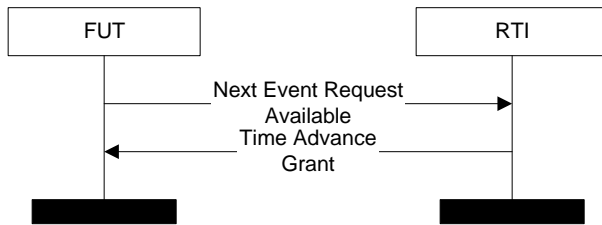
#### 3.5.6.2.2 Next Event Request Available

The sequence represents the service invocation required for a federate to execute Next Event Request Available.

##### 3.5.6.2.2.1 Traceability

Section 8.11 Next Event Request Available and Section 8.13 Time Advance Grant †.

#### 3.5.6.2.2.2 Service Set



The *Next Event Request Available* service shall request the logical time of the federate to be advanced to the time stamp of the next TSO message that will be delivered to the federate, provided that message has a time stamp no greater than the logical time specified in the request.

Invocation of the *Time Advance Grant* † service shall indicate that a prior request to advance the federate's logical time has been honored.

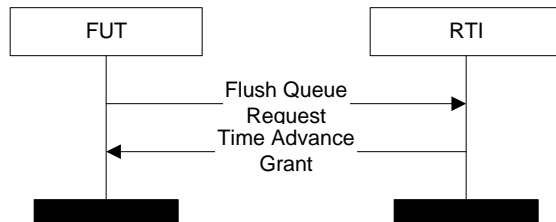
### 3.5.6.3 Flush Queue Request

The sequence represents the service invocation required for a federate to execute Flush Queue Request.

#### 3.5.6.3.1 Traceability

Section 8.12 Flush Queue Request and Section 8.13 Time Advance Grant †.

#### 3.5.6.3.2 Service Set

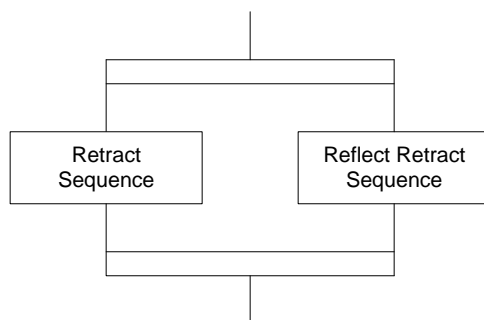


The *Flush Queue Request* service shall request that all messages queued in the RTI that the federate will receive as TSO messages be delivered now.

Invocation of the *Time Advance Grant* † service shall indicate that a prior request to advance the federate's logical time has been honored.

### 3.5.7 Retract Sequences

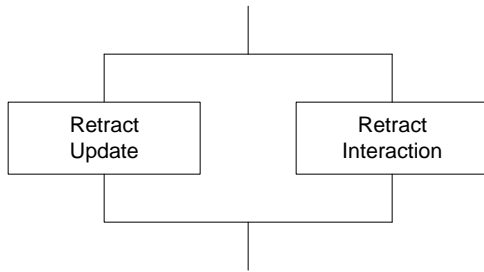
The Retract sequences represent the service sets associated with Retracting Attribute Updates and Interactions. The high-level MSC is shown below. These sequences are represented in a PARALLEL MSC since they can be executed in any order.



#### 3.5.7.1 Retract Sequence

The Retract sequences represent the service sets associated with retracting Updates and Interactions. The high-level MSC is shown below. These sequences are represented in an OR MSC since either an Update or Interaction would be retracted at one time.





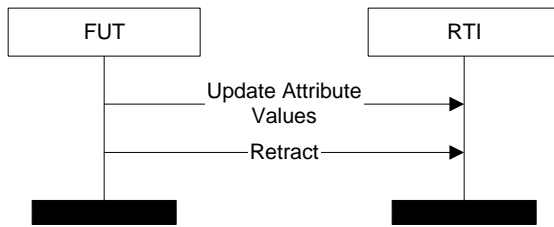
### 3.5.7.1.1 Retract Update

The sequence represents the service invocation required for a federate to Retract Update.

#### 3.5.7.1.1.1 Traceability

Section 6.4 Update Attribute Values and Section 8.21 Retract.

#### 3.5.7.1.1.2 Service Set



The *Update Attribute Values* service shall provide current values to the federation for instance attributes owned by the federate.

The *Retract* service shall be used by a federate to notify the federation execution that a message/event previously sent by the federate is to be retracted.

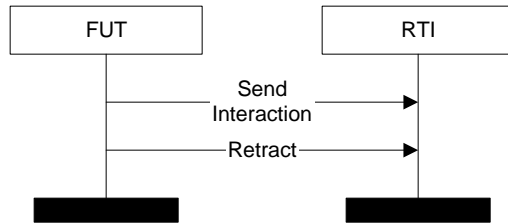
### 3.5.7.1.2 Retract Interaction

The sequence represents the service invocation required for a federate to Retract Interaction.

#### 3.5.7.1.2.1 Traceability

Section 6.6 Send Interaction and Section 8.21 Retract.

#### 3.5.7.1.2.2 Service Set

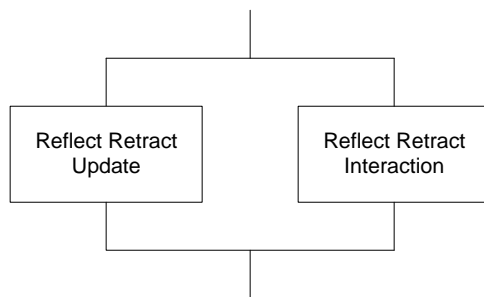


The *Send Interaction* service shall send an interaction into the federation.

The *Retract* service shall be used by a federate to notify the federation execution that a message/event previously sent by the federate is to be retracted.

### 3.5.7.2 Reflect Retract Sequences

The Reflect Retract sequences represent the service sets associated with retracting Reflected Updates and Interactions. The high-level MSC is shown below. These sequences are represented in an OR MSC since either an Update or Interaction would be retracted at one time.



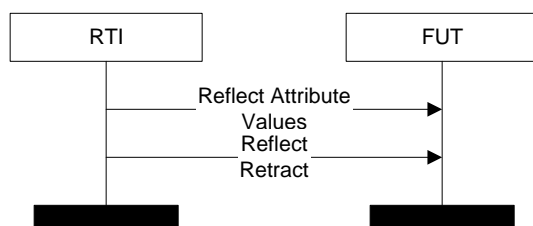
#### 3.5.7.2.1 Reflect Retract Update

The sequence represents the service invocation required for a federate to Retract a Reflected Update.

##### 3.5.7.2.1.1 Traceability

Section 6.5 Reflect Attribute Values † and Section 8.22 Request Retraction †.

##### 3.5.7.2.1.2 Service Set



The *Reflect Attribute Values* † service shall provide the federate with new values for the specified instance attributes.

If the RTI receives a legal *Retract* service invocation for an event that has already been delivered to a federate, the *Request Retraction* † service shall be invoked on that federate.

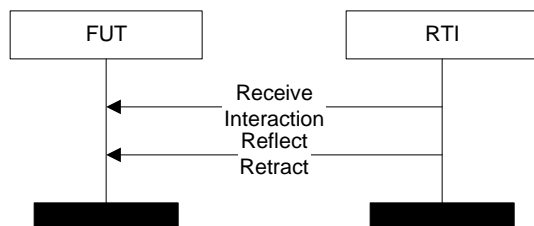
#### 3.5.7.2.2 Reflect Retract Interaction

The sequence represents the service invocation required for a federate to Retract a Reflected Interaction.

##### 3.5.7.2.2.1 Traceability

Section 6.7 Receive Interaction † and Section 8.22 Request Retraction †.

##### 3.5.7.2.2.2 Service Set

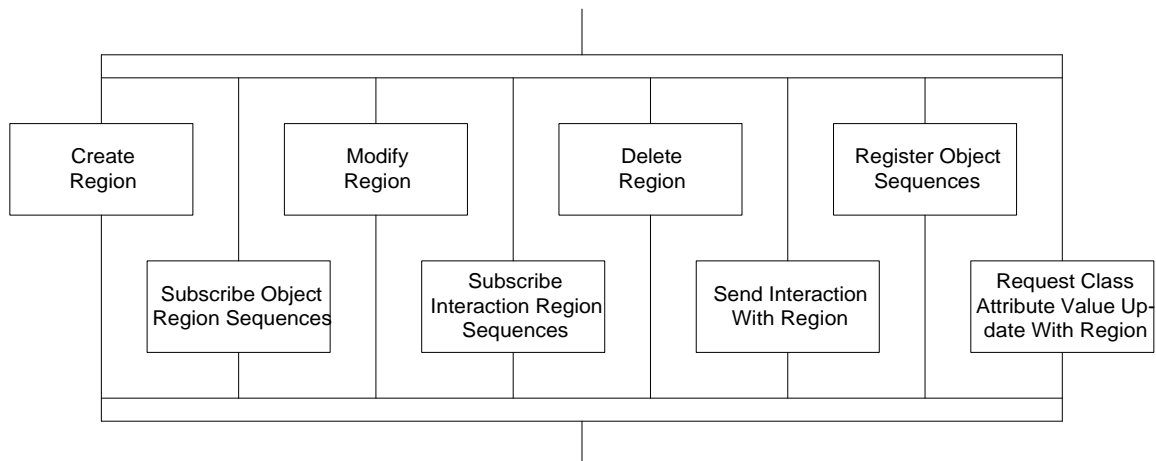


The *Receive Interaction* † service shall provide the federate with a sent interaction.

If the RTI receives a legal *Retract* service invocation for an event that has already been delivered to a federate, the *Request Retraction* † service shall be invoked on that federate.

### 3.6 DATA DISTRIBUTION MANAGEMENT SEQUENCES

The Data Distribution Management (DDM) Sequences represent the service sets associated with DDM. These service sequences are shown in the high-level MSC below.



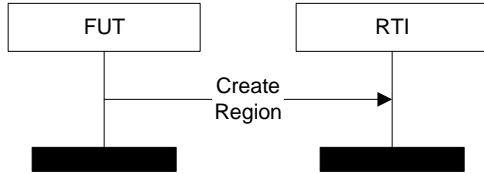
#### 3.6.1 Create Region

This sequence represents the service invocation required for a federate to Create Region.

#### 3.6.1.1 Traceability

Section 9.2 Create Region.

#### 3.6.1.2 Service Set



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

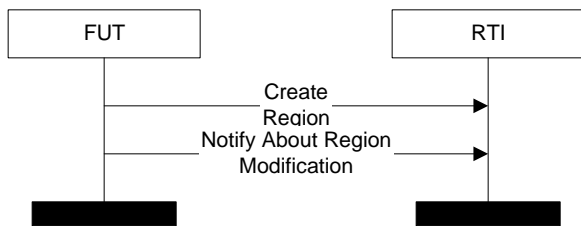
### 3.6.2 Modify Region

This sequence represents the service invocation required for a federate to Modify Region.

#### 3.6.2.1 Traceability

Section 9.2 Create Region and Section 9.3 Modify Region.

#### 3.6.2.2 Service Set



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The *Modify Region* service shall inform the RTI about changes to the extent set of the region.

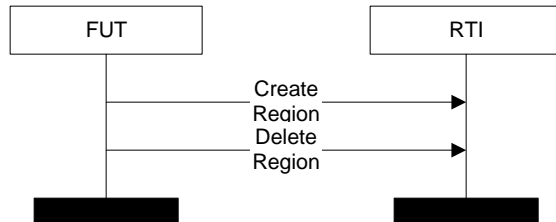
### 3.6.3 Delete Region

This sequence represents the service invocation required for a federate to Delete Region.

#### 3.6.3.1 Traceability

Section 9.2 Create Region and Section 9.4 Delete Region.

#### 3.6.3.2 Service Set

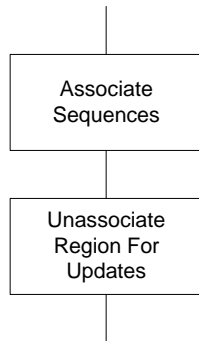


The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The *Delete Region* service shall delete the specified region.

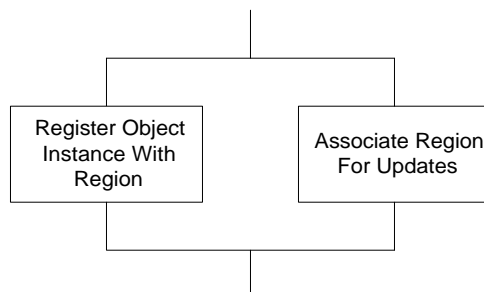
### 3.6.4 Register Object Sequences

The Register Object sequences represent the service sets associated with Registering Object Attributes with a Region. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.



#### 3.6.4.1 Associate Sequences

The Associate sequences represent the service sets associated with associating attributes with a region. These sequences are shown below in an OR MSC since they represent alternatives.



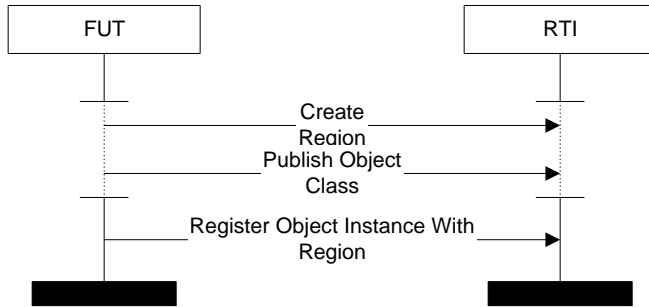
##### 3.6.4.1.1 Register Object Instance with Region

This sequence represents the service invocations required for a federate to Register Object Instance with Region. The hash marks along the FUT and RTI axes in the MSC represent co-regions in which service invocations can be executed in any order.

#### 3.6.4.1.1.1 *Traceability*

Section 9.2 Create Region, Section 5.2 Publish Object Class, and Section 9.5 Register Object Instance With Region.

#### 3.6.4.1.1.2 *Service Set*



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The information conveyed by the federate via the *Publish Object Class* service shall be used in multiple ways. First, it shall indicate an object class of which the federate may subsequently register object instances. Second, it shall indicate the class attributes of the object class for which the federate is capable of owning the corresponding instance attributes of object instances whose known class is that class.

The *Register Object Instance With Region* service shall create a unique object instance designator and link it with an object instance of the supplied class.

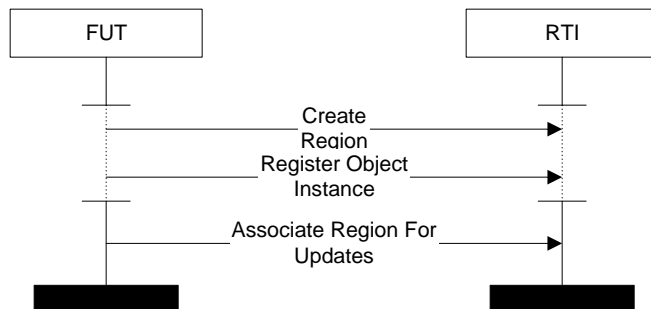
#### 3.6.4.1.2 *Associate Region for Updates*

This sequence represents the service invocations required for a federate to Associate Region for Updates. The hash marks along the FUT and RTI axes in the MSC represent co-regions in which service invocations can be executed in any order.

#### 3.6.4.1.2.1 *Traceability*

Section 9.2 Create Region, Section 6.2 Register Object Instance, and Section 9.6 Associate Region For Updates.

#### 3.6.4.1.2.2 *Service Set*



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The RTI shall create a unique object instance designator and shall link it with an instance of the supplied object class.

The *Associate Region For Updates* service shall associate a region to be used for updates with instance attributes of a specific object instance.

#### 3.6.4.2 Unassociate Region for Updates

This sequence represents the service invocation required for a federate to Unassociate Region for Update.

##### 3.6.4.2.1 Traceability

Section 9.7 Unassociate Region For Updates.

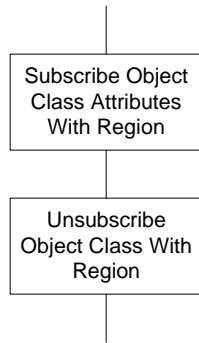
##### 3.6.4.2.2 Service Set



The *Unassociate Region For Updates* service shall remove the association between the region and all instance attributes associated with that region.

### 3.6.5 Subscribe Object Region Sequences

The Subscribe Object Region sequences represent the service sets associated with Subscribing objects with a region. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.



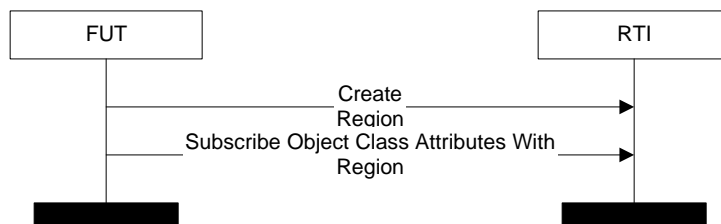
### 3.6.5.1 *Subscribe Object Class Attributes with Region*

This sequence represents the service invocations required for a federate to Subscribe Object Class with Region.

#### 3.6.5.1.1 Traceability

Section 9.2 Create Region and Section 9.8 Subscribe Object Class Attributes With Region.

#### 3.6.5.1.2 Service Set



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The *Subscribe Object Class Attributes With Region* service shall specify an object class for which the RTI is to begin notifying the federate of discovery of instantiated object instances when at least one of that instance's attributes are in scope.

### 3.6.5.2 *Unsubscribe Object Class with Region*

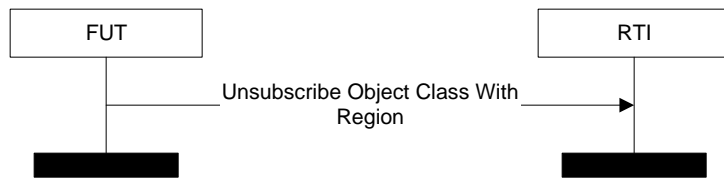
This sequence represents the service invocation required for a federate to Unsubscribe Object Class with Region.

#### 3.6.5.2.1 Traceability

Section 9.9 Unsubscribe Object Class With Region.

#### 3.6.5.2.2 Service Set

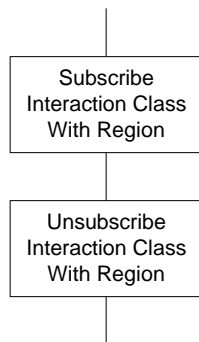




The Unsubscribe Object Class With Region service shall inform the RTI that it shall stop notifying the federate of object instance discoveries for the specified object class.

### 3.6.6 Subscribe Interaction Region Sequences

The Subscribe Interaction Region sequences represent the service sets associated with Subscribing an Interaction Class to a Region. The high-level MSC is shown below. These sequences are represented in an AND MSC since they must be executed in the specified order.



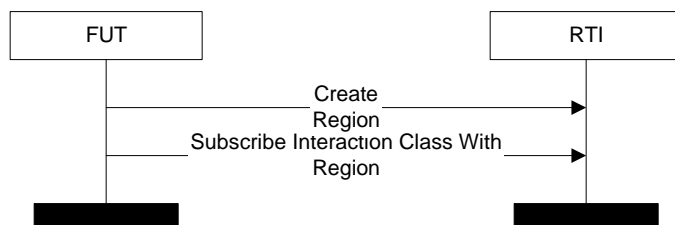
#### 3.6.6.1 *Subscribe Interaction Class with Region*

This sequence represents the service invocations required for a federate to Subscribe Interaction Class with Region.

##### 3.6.6.1.1 Traceability

Section 9.2 Create Region and Section 9.10 Subscribe Interaction Class With Region.

##### 3.6.6.1.2 Service Set



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The Subscribe Interaction Class With Region service shall specify the class of interactions that should be delivered to the federate, taking the region into account.

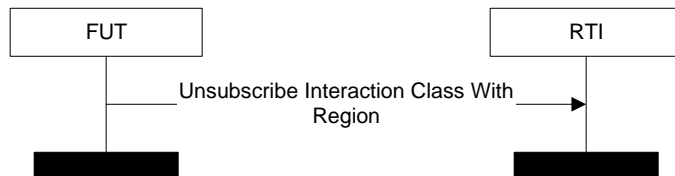
### 3.6.6.2 *Unsubscribe Interaction Class with Region*

This sequence represents the service invocation required for a federate to Unsubscribe Interaction Class with Region.

#### 3.6.6.2.1 Traceability

Section 9.11 Unsubscribe Interaction Class With Region.

#### 3.6.6.2.2 Service Set



The *Unsubscribe Interaction Class With Region* service shall inform the RTI no longer to notify the federate of sent interactions of the specified class that are in the specified region.

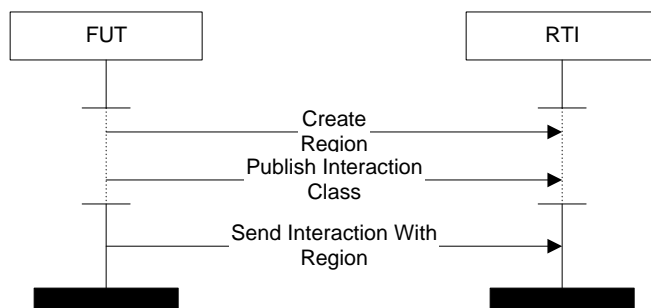
### 3.6.7 **Send Interaction with Region**

This sequence represents the service invocations required for a federate to Send Interaction with Region. The hash marks along the FUT and RTI axes in the MSC represent co-regions in which service invocations can be executed in any order.

#### 3.6.7.1 *Traceability*

Section 5.4 Publish Interaction Class, Section 9.2 Create Region, and Section 9.12 Send Interaction With Region.

#### 3.6.7.2 *Service Set*



The *Create Region* service shall create a region that has the dimensions of the specified routing space and the specified number of extents.

The *Publish Interaction Class* service shall inform the RTI which classes of interactions the federate will send to the federation execution.

The *Send Interaction With Region* service shall send an interaction into the federation.

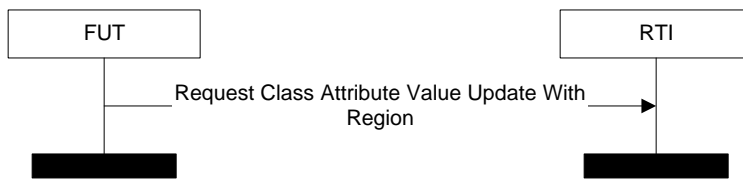
### 3.6.8 Request Class Attribute Value Update with Region

This sequence represents the service invocation required for a federate to Request Class Attribute Value Update with Region.

#### 3.6.8.1 Traceability

Section 9.13 Request Attribute Value Update With Region.

#### 3.6.8.2 Service Set



The *Request Attribute Value Update With Region* service shall be used to stimulate the update of values of specified attributes.

#### **4. REFERENCES**

- [1] "HLA Compliance Checklist, v1.3," Alexandria, VA: Defense Modeling and Simulation Office, April 1998.
- [2] "HLA Conformance Guide, v1.3," Alexandria, VA: Defense Modeling and Simulation Office, April 1998.
- [3] E. Rudolph, J. Grabowski, and P. Graubmann, "Tutorial on Message Sequence Charts (MSC 96)," presented at Tutorial of the FORTE/PSTV'96, Kaiserslautern, Germany, 1996.
- [4] K. Knightson, OSI Protocol Conformance Testing: IS 9646 Explained, McGraw-Hill, Inc., 1993.